

Джим Хаганин » Opera 9 » OSDL » Xfce 4.4
LXF DVD внутри!
LINUX FORMAT
ГЛАВНОЕ В МИРЕ LINUX
октябрь 2006 № 10 (84)
SimplyMepis 6
Knoppix 5
Плюс: еще 3 дистрибутива, свежий Mono, OOo и другие новинки
с. 112 »
AMD vs Intel
Виртуальный прорыв!
Новые процессоры меняют правила виртуализации
» Результаты тестирования
» Обзор технологии
Плюс: Пошаговое руководство по совместному использованию Intel VT и Xen
Google Earth
Linux-версия вышла на орбиту – время тестировать!
Что для нас сделала OSDL?
Тайны самой важной Linux-организации
НОВАЯ СЕРИЯ
Безопасность
Что делает пароль паролем?
Когда я впервые увидел Java, то подумал, что это какая-то шутка
Джим Хаганин с. 32
Linux center
www.linuxcenter.ru
4 607130 820445



К вашим услугам

Глядя на новенький с иголочки *Linux Format*, мы решили поинтересоваться у его команды: А какое ПО для Linux более всего нуждается в переработке?



Пол Хадсон
Emacs. Режим Viper необходимо включить по умолчанию!



Грэм Моррисон
Есть такой маленький проект *Kalbum* – ему очень не хватает внимания.



Майк Сондерс
Команда *ls*. Как же так – без 3D-интерфейса, без текстур, без скрытых функций...



Эфрейн Эрнандес-Мендоса
Да, в общем, все... Пингвинов маловато.



Ребекка Смоли
Gimp звучит вяло. Переименовать бы его в *DreamPhotos* или *Imageine* и украсить портретами кинозвезд.



Эндрью Грегори
Я хочу, чтобы дух Отто Вагнера занялся архитектурой *Yast* – знатно бы вышло.



Марк Бейн
Никакое – но я по-прежнему уверен, что *XPilot* – это круто!.



Нейл Ботвик
Я бы предложил Gnome, но, боюсь, приятный внешний вид озадачит его пользователей...



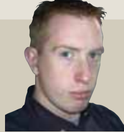
Д-р Крис Браун
dd. Ее параметры – вопиющее нарушение стандартов Unix. Правда, я бы предпочел лечению эвтаназию.



Дэвид Картрайт
X.
Они моложе меня, но гораздо толще, капризнее и менее дружелюбны.



Энди Ченнел
OOo нужно как-то охладить пыл пользователей. Как насчет чего-нибудь жутко сложного в освоении?



Дэвид Кулсон
Я с трудом гляжу в окно *GnomeMeeting*, но, кажется, проблема не в нем, а в том, на кого направлена web-камера.

Миссия журнала

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО

Corporate Patron Free Software Foundation



Совсем другая тема?

Все, как известно, любят темы. С легкой руки законодателя мод компьютерного мира – компании Apple, эта тенденция распространилась на мультимедийные проигрыватели, web-браузеры, целые операционные системы, сотовые телефоны (как изнутри, так и снаружи)... и вот, наконец, добралась и до нашего журнала.

Первое, что должно бросаться вам в глаза, едва вы глянете на эту страницу – новое фото главного редактора. Умело откадрированный групповой снимок с Ларри Уоллом из прошлых выпусков уступил место более солидному и широкоформатному образцу репортажного фото. Благодаря этому нововведению, у меня стало чуть меньше место для изложения своих мыслей на данной странице, поэтому, видимо, надо потихоньку закругляться, подобно новому дизайну журнала...

На его смену ушло, кстати, около полугода упорного труда. Надеюсь, что вам понравится результат, хотя, конечно, найдутся читатели, предпочитающие старый вариант оформления. Не беспокойтесь – к хорошему быстро привыкаешь, да и за новой цветной обложкой скрывается ваш старый знакомый. **LXF**

Валентин Синицын » Главный редактор info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Website: www.linuxformat.ru
› **Адрес редакции:** Россия, Санкт-Петербург, ул. Гончарная, 23, офис 54.
› **Телефон редакции:** (812) 717-00-37. Дополнительная информация на стр.118



октябрь 2006 № 10 (84)

Содержание

Весь номер – прямо как на ладони: приятного чтения!

Учебники





LXF DVD84

SimplyMepis 6.0
Легко установить – легко использовать: как поживается этому дистрибутиву на плечах Ubuntu? Плюс: руководство по инсталляции

Knoppix 5.0
Последняя версия первого LiveCD. Читайте руководство на DVD!

PC-BSD 1.2
Скорость и стабильность FreeBSD с приятным рабочим столом. Если вы никогда не пробовали FreeBSD, обратите внимание.

Zenwalk Linux
Разработчики обкорнали Slackware и получили быстрый, компактный, приятный дистрибутив.



› Запустите Live-версию Mepis или инсталлируйте ее на свой винчестер.

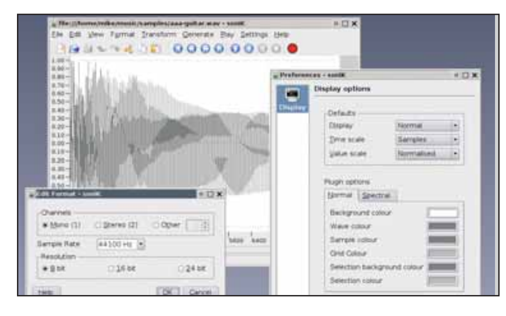
Что за штука...
Java Server Faces?

Web-программисты и дизайнеры могут работать вместе
с. 46

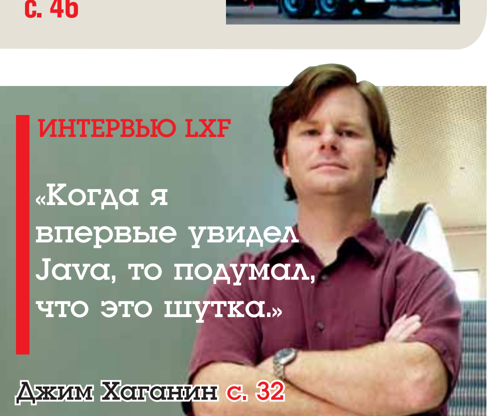




LXFHotPicks



› *Sonik*-бум в мире аудиософта под Linux...



ИНТЕРВЬЮ LXF

«Когда я впервые увидел Java, то подумал, что это шутка.»

Джим Хаганин с. 32



Усидеть на нескольких стульях с. 38

Амбиции The Open Source Development Labs



Подпишись на Linux Format и сэкономь!

подробности на стр.119



LXF DVD внутри!

См. страницу 112



Спецрепортаж

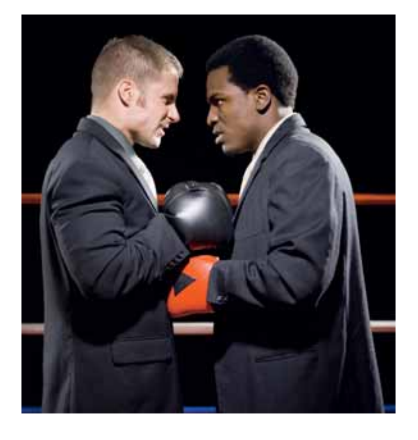
Виртуальный прорыв

Новые процессоры меняют представление о виртуализации с. 22



А также...

Постоянные рубрики



Обзоры





Python 2.5



Новая версия популярного языка сценариев имеет все шансы стать еще популярнее. Почему – разбирается **Сергей Супрунов**.

19 сентября 2006 года, после нескольких месяцев бета-тестирования и двух кандидатов в релизы, увидел свет финальный релиз Python 2.5. Среди основных улучшений – увеличение скорости работы, особенно на операциях со строками и при обработке исключений; улучшенная поддержка Unicode; новые операторы языка, призванные сделать его ещё более удобным; многочисленные дополнения и улучшения во входящих в дистрибутив модулях.

Среди новых языковых средств можно упомянуть:

» **Новый условный оператор**. Теперь вы можете писать var = A if <условие> else B, то есть если <условие> истинно, то переменная var получит значение A, иначе – B. Это напоминает тернарный оператор <условие> ? A : B, но ближе к естественному языку, хотя и не совсем привычно.

» **Единый оператор** try-except-finally. В прежних версиях использовалось два оператора для обработки исключений: try-except и try-finally. Первый позволял обработать возникшую ошибку в секции except, второй гарантировал выполнение кода в секции finally даже в случае возникновения критической ошибки в секции try (это, например, часто используется для гарантированного закрытия открытых файл). Теперь эти функции объединены в одном операторе:

```
fd = open('file.txt', 'w')
try:
    fd.write('Hello')
except:
    raise 'Ошибка записи'
finally:
    fd.close()
```

» **Оператор** with. Впрочем, вместо try-finally теперь можно использовать новый оператор with

```
with open('file.txt', 'w') as fd:
    fd.write('Hello')
```

В данном случае дескриптор fd будет существовать только в пределах блока with, и при выходе из блока (в том числе и в результате исключения) файл будет гарантированно закрыт.

» **Новый метод словаря**. Тип данных словарь получил ещё один встроенный метод: __missing__(). Он будет вызываться, если выполняется обращение к словарю по отсутствующему ключу, и следовательно, программист получает возможность переопределить этот метод в производных классах. В предыдущих версиях Python отсутствие ключа в словаре приводило к исключению KeyError (по умолчанию это поведение сохранилось).

» **Параметр** key. Функции min() и max(), по примеру функции sort(), обзавелись дополнительным параметром key. Теперь можно указать свою функцию сравнения. Например, max(('abc', 'bc', 'c'), key=len) 33 вернёт значение 'abc', как имеющее самую большую длину. Без параметра key будет возвращено 'c' как самое «большое» по алфавиту.

Ряд изменений внесён в модуль os. В частности, функция os.stat() по умолчанию теперь будет выводить значения дат в виде чисел с плавающей запятой. Добавлена пара новых функций – wait3() и wait4(), позволяющих более гибко реализовывать ожидание завершения дочерних процессов.

Модуль socket теперь поддерживает и Linux-сокеты AF_NETLINK, используемые для коммуникации между пользовательскими процессами и ядром в Linux.

Модуль threading позволяет устанавливать размер стека для вновь создаваемых потоков, для чего используется функция stack_size([size]).

Добавлен ряд новых модулей, среди которых: uuid, позволяющий генерировать уникальные идентификаторы; sqlite3 для работы с одноимённой БД; ElementTree для обработки XML-данных, и ряд других.

Окончательно удалены устаревшие модули regex и regsub, а также ряд других.

Этим перечень изменений не ограничивается – всего отмечено около 800 патчей и исправлений ошибок, так что можно сказать, что Python «стал лучше во всех отношениях».
www.python.org

Что было

SofTool'06/LinuxLand

26-29 сентября 2006 года

В рамках ежегодной выставки информационных технологий *SofTool* в Москве прошла вторая по счету экспозиция LinuxLand и конференция «ИТО-2006: Технологии Linux и Open Source». Участники конференции имели возможность прослушать доклады о внедрении Linux в странах СНГ, а также поучаствовать в тренингах по современным открытым технологиям. Кроме этого, в рамках *LinuxLand* состоялось награждение победителей конкурса IBM WebSphere Community Edition Contest.
См. обзор LinuxLand в LXF85.
› www.linuxland.ru

Gnome 2.16



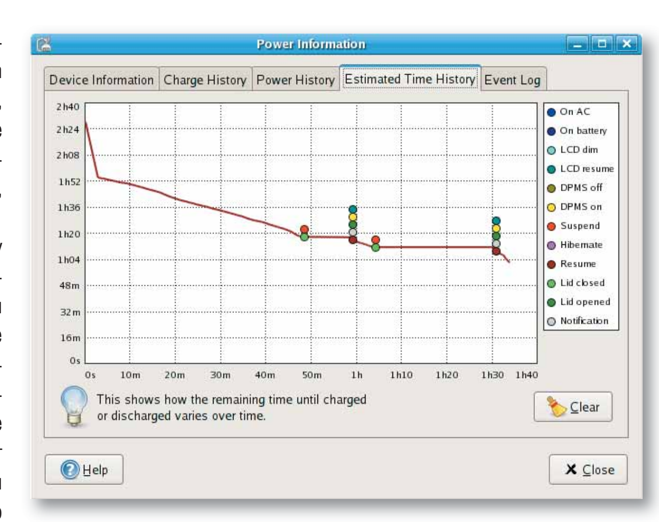




Изменился и по форме, и по содержанию.

В начале осени сторонники рабочей среды *Gnome* (коих в сообществе Open Source, согласно некоторым оценкам, никак не меньше трети) отметили выход ее новой версии – *2.16*. Многочисленные изменения коснулись как чисто визуальной стороны, так и функционала. Но – обо все по порядку.

Пользователи ноутбуков по достоинству оценят новый энергосберегающий инструмент *Gnome*. Помимо стандартных функций по уменьшению яркости экрана, установке времени «засыпания» и т.п., теперь доступен график зависимости оставшегося времени автономной работы от времени (простите за невольный каламбур), который позволяет наглядно видеть, насколько эффективны ваши меры по сохранению энергии. Пока компьютер находится в «спячке», зависимость практически горизонтальная, а во время «бодрствования» наблюдаются резкие падения.



› Новый встроенный инструмент управления питанием.

Довольно об оборудовании. Все-таки, компьютер – это лишь инструмент, призванный автоматизировать рутинные или малоинтересные операции, например, составление напоминаний самому себе. Радуйтесь – в состав *Gnome 2.16* включен *Tomboy*, система экранных заметок, основанная на Mono. Tomboy не раз получал высокую оценку редакции LXF, и мы наконец-то рады видеть его в стандартной поставке.

Чтобы создать новую заметку, достаточно щелкнуть мышью в любом месте рабочего стола и ввести текст в появившемся окне (имеется встроенная проверка орфографии). Заметки также можно связывать между собой, только не увлекайтесь этим чрезмерно – все-таки, это не Wiki, а ваш рабочий стол – еще не весь Интернет.

Редактировать меню *Gnome* тоже стало значительно проще благодаря введению в строй нового инструмента – *Alacarte*. Пользователи Ubuntu должны хорошо знать эту маленькую утилиту, а теперь она стала доступна и для людей, предпочитающих другие дистрибутивы.

Однако, никакой, даже самый развитый функционал не может удовлетворить тягу к прекрасному – и *Gnome* стремится тщательно следовать модным тенденциям. Версия *2.16* является в этом смысле переходной – многие из заложенных в ней нововведений проявятся в полной мере лишь в *Gnome 2.18*, но и сейчас владельцам мощных видеокарт уже есть, чем нагрузить свои GPU. Речь, разумеется, идет от трехмерных эффектах. Благодаря изменениям в Metacity, окна *Gnome* теперь умеют дрожать, сжиматься, взрываться, плавно исчезать и еще много чего другого. Единственное «но», помимо необходимости иметь приличную видеокарту (пока поддерживаются только Intel i830-i945 и ATI Radeon 7000-9250), состоит в требовании скомпилировать Metacity с ключом --enable-compositor – не все сборщики пакетов пожелают связываться с экспериментальной функцией.

Gnome 2.16 будет рабочим столом по умолчанию в Fedora Core 6 и Ubuntu 6.10 так что, к тому времени, когда вы будете читать эти строки, новинку уже вполне можно будет попробовать в деле.

www.Gnome.org

P.S. Читайте обзор Gnom2 2.16 в LXF85





Latest News

GNOME 2.16 Released!

The GNOME desktop is now faster and easier to configure than ever before. Learn more about this release through the GNOME 2.16 start page and the detailed release notes.

Women's Summer Outreach Program 2006

Following on from GNOME's participation in Google's Summer of Code, we've decided to sponsor three projects in a similar fashion to the Summer of Code, but for women only. GNOME had no Summer of Code applications from women, and we think it's time to do something to encourage more women to join our

Upcoming Events

Boston Summit 2006

7-9th October, 2006; Boston (USA)

Three-day hackfest for GNOME developers and contributors. Meet the developers. Share your ideas. Get involved in GNOME's present and future.

What is GNOME?

GNOME offers an easy to understand desktop for your Linux or UNIX computer. Learn more about GNOME as a whole, get our latest release, find help and support, join our great community, start to develop with GNOME, study our Foundation, or simply get in contact with us.

GNOME Shirts Available!

Hackerthreads will be donating a portion of proceeds of the shirt sales to the GNOME Foundation. Buy a shirt today!

Новости короткой строкой

» Компания Mandriva обновила свою линейку дистрибутивов, выпустив Mandriva Linux 2007 и Mandriva Corporate Server 4.0

» Novell готовится готовится к выпуску системы реального времени SUSE Linux Enterprise Real-Time

» В Виртуальной Библиотеке ЛинуксЦентра опубликован ответ на статью Владимира Попова «Строим сеть с терминальным сервером сами» (LXF81)**http://linuxcenter.ru/lib/articles/soft/x_terminals2.phtml**

» Увидели свет: *Subversion 1.4.0*, *Gwenview 1.4.0*, Linux 2.6.18

Обзор выставки LinuxWorld-2006

Андрей Маркелов

С четвертого по пятое сентября в Москве, в Центральном выставочном комплексе «Экспоцентр» прошла уже вторая по счету Российская выставка-конференция *LinuxWorld*. Нужно сказать, что организация самого мероприятия несколько отличалась от прошлогодней выставки не только местом проведения, но и еще одним сразу бросающимся в глаза моментом. Так же, как и в прошлом году, вместе с *LinuxWorld* одновременно проходили Storage Expo, Infosecurity и Documation. Но, в отличие от прошлогодней выставки, где стенды Linux-компаний располагались на общем для всех четырех мероприятий пространстве, на этой выставке *LinuxWorld* был вынесен в отдельный уголок Экспоцентра. К сожалению, это произошло не из-за увеличившегося количества участников. Скорее даже наоборот – ряда компаний, которые мы видели в 2005 году, на новой выставке не было. И как признали те участники выставки, с которыми я общался, такое выделение из «общего котла» выставляло *LinuxWorld 2006* не с лучшей стороны, подчеркнув как относительно небольшое число участников, так и зачастую пустые стенды-коробки, на которых можно было взглянуть разве что на пару-другую представителей компаний.

Но не будем о грустном, а лучше скажем о том, что понравилось. Конечно же, подобные выставки в первую очередь – это общение. Компании могут рассказать о предлагаемых услугах и продуктах. Потенциальные заказчики – задать интересующие вопросы и познакомиться с предложениями на примере того среза рынка, который представлен на выставке. Ну и, наконец, подобные выставки – это место, где профессионалы могут пообщаться «вживую», что называется, глаза в глаза с собеседником, а не со всплывающими окошками IM или интерфейсом почтового клиента. Не будем забывать и о тематических семинарах, собиравших достаточно большое число слушателей.

Переходя же к конкретике, среди компаний, отметившихся на *LinuxWorld,* можно выделить нескольких представителей Linux-рынка России. Безусловно, одну из первых скрипок играла компания **IBM**, которая не только являлась платиновым спонсором конференции, но и представила самые большие по площади стенды. Параллельно с основными секциями конференции, все на том же стенде IBM проходили собственные презентации и доклады компании. Правда, в тот момент, когда я прогуливался мимо стенда, большой плазменный экран демонстрировал не красоты рабочего стола SLES (а надо сказать, что 3D-переключение рабочих столов в последней версии этого корпоративного дистрибутива действительно реализовано впечатляюще) или RHEL, а слайды в обрамлении интерфейса *Microsoft PowerPoint*.

Еще одной точкой, по традиции вызвавшей заметное скопление заинтересованной публики, как и на всех подобных выставках, оказался стенд компании **LinuxCenter.Ru**. Это и не удивительно, учитывая что там можно было приобрести множество различных дистрибутивов Linux, атрибутику и специализированную литературу, в том числе и журнал **Linux Format** за последние несколько месяцев.

Кстати, говоря о периодике, нельзя не упомянуть и стенд журнала «Системный администратор», где все желающие могли не только пообщаться с представителями редакции, но и обменять денежные знаки на журналы по льготному курсу.

На выставке были также представлены и компании-производители дистрибутивов. Были представлены две самые известные отечественные компании, выпускающие свои дистрибутивы – ASP Linux и ALT Linux, в то время как из зарубежных компаний-лидеров корпоративного рынка Linux-систем, разрабатывающих собственные дистрибутивы, участвовала только Novell.

Из других компаний, участвовавших в выставке, хотелось бы упомянуть **Linux Ink** (являвшуюся, помимо прочего, организатором конференции), **R-Style**, **MSI**, **Softline**, **ОАО «ВНИИС»**, издательство **«Открытые системы»**, **Центр верификации ОС Linux**.

Достаточно большой интерес у посетителей вызвали семинары и доклады, проходившие на выставке. Сама конференция была разбита на несколько параллельно идущих секций: бизнес-секция, техническая секция и семинары компаний. К сожалению, охватить все доклады не было совершенно никакой возможности: в частности, я так и не смог побывать на семинаре компании Microsoft, сотрудники которой обещали поделиться взглядами на бизнес-модель, основанную на открытом коде, и на взаимодействие с сообществом разработчиков. Из тех докладов, на которых удалось побывать, хотелось бы отметить серию интересных выступлений, посвященных новой версии корпоративного дистрибутива SUSE Linux Enterprise 10 от Novell, и любопытный рассказ о применении SELinux в проекте для правительства Великобритании. Также достаточно интересно было узнать о завершающейся сертификации RHEL на соответствие российским стандартам. Правда, интересна эта новость будет лишь клиентам IBM, т.к. процедура сертификации привязывается не только к конкретной версии дистрибутива и пакету обновлений, но и к конкретному оборудованию.



› **Выставочные стенды компаний-участников LinuxWorld-2006.**

В конце этого небольшого обзора хотелось бы выразить надежду, что на *LinuxWorld-2007* мы увидим новые компании-участники, еще больше интересных докладов, и познакомимся с большим числом решений, представленных на рынке Linux-систем. LXF





Обзоры



Новинки программного и аппаратного обеспечения в описании наших экспертов



АЛЕКСЕЙ ФЕДОРЧУК
Свою первую (и последнюю) программу написал еще на Алголе.

LinuxWorld 2006

Может быть, кое-кто из моих читателей помнит чувство сопричастности высоким технологиям, охватывавшее посетителей первых компьютерных выставок в конце 80-х – начале 90-х годов прошлого тысячелетия. Или – первые выставки UnixExpo середины 90-х – когда нашим глазам впервые предстала вся мощь рабочих станций, оснащенных разными вариантами одноименной операционной системы. Ну и, наконец, многим памятен прошлогодний каскад выставок, посвященных Open Source и Linux, оставивший ощущение того, что эти сферы перестали быть периферией IT-мира.

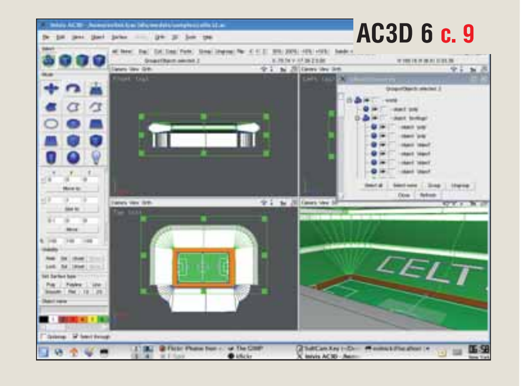
> «Мир Linux и Open Source окончательно миновал стадию ажиотажного развития и вступил в период планомерного практического применения.»

Традиция выставок продолжена была и в текущем году. Вслед за вторым Open Source Forum Russia настало время и LinuxWorld (4–5 сентября). Первое впечатление от нее – будничность. Немногочисленные посетители степенно беседуют с представителями экспонирующихся фирм. Что же до последних – ба, знакомые все лица, в очередном перечислении не нуждающиеся. Не показатель ли это того, что мир Linux и Open Source окончательно миновал стадию ажиотажного развития и вступил в период планомерного практического применения? Думаю, так – и это не может не радовать. Но, с другой стороны, в жизни должно быть место празднику. Может быть, им станет грядущий LinuxLand? В момент, когда вы будете читать эти строки, ответ будет получен...

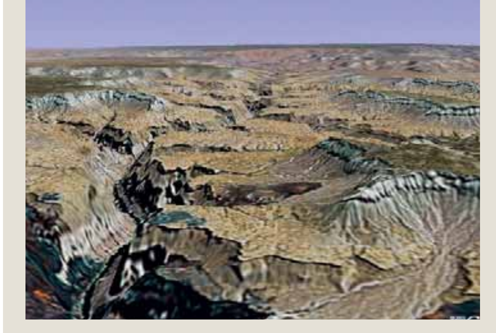
alv@posix.ru

Сегодня мы рассматриваем...

AC3D 6 с. 9



Google Earth с. 8



› То березка, то рябина, куст ракиты над рекой..

Gorky 17 с. 12



› Кошелек или жизнь? Хотя, с такой мордой как у тебя... кошелек и жизнь!

НАШ ВЕРДИКТ пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют Autopackage, мы следуем этому совету.



LINUX FORMAT Вердикт

Google Earth

Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» *Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная*

Рейтинг 9/10

Google Earth



Предвидя захват Земли хакерами, **Грэм Моррисон** рассматривает проект Google – никогда ещё виртуальная жизнь не была так хороша.

Вкратце...

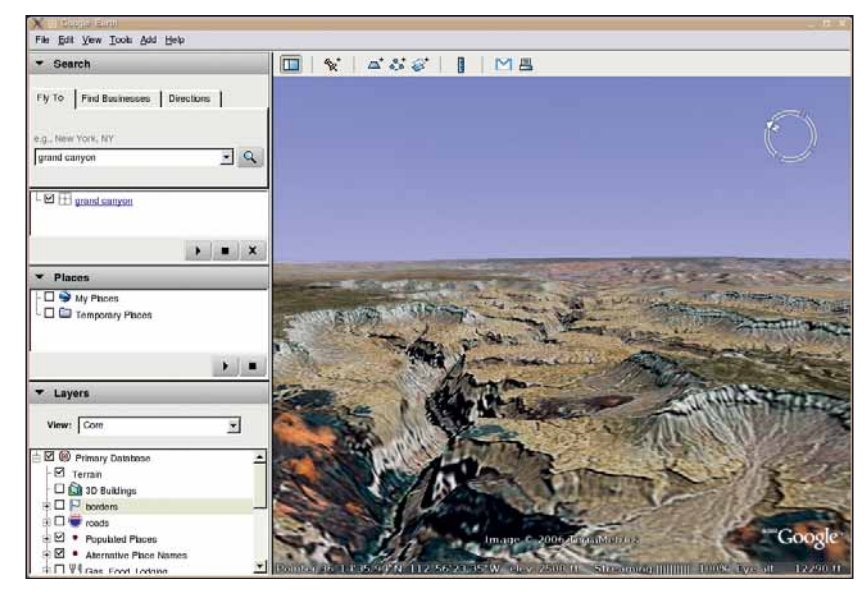
» Исследуйте Землю за своим рабочим столом с помощью спутниковых изображений и 3D-данных. См. также: Celestia или ожидаемый NASA World Wind 1.5.

Надо признать, что год назад, когда был выпущен *Google Earth* для Windows, впечатление было сильным. Мы завистливо наблюдали, как знакомые почитатели Windows и Mac запускают виртуальную планету Google и бороздят Тихий океан в поисках неисследованных территорий или просто удобного места для отпуска. Но больше мы им не завидуем! Google совершил благородный поступок и выпустил родной порт для старого доброго Linux. Родной – это вам не полуподпольные библиотеки *Wine* для обработки системных вызовов Windows, как в *Picasa* (*см. обзор LXF83*). Всё сделано основательно, с помощью *Qt* и под OpenGL.

Для тех, кому пока не посчастливилось наблюдать *Google Earth* в действии – это естественное расширение Google Maps. При первом запуске мы видим голубое пятнышко Земли с расстояния порядка 15 000 миль. Прокрутка мыши приближает нас к родной планете, пока мы не зависнем над США. Чем дольше вы держитесь на одном месте, тем детальнее становится вид – *Google Earth* безостановочно транслирует спутниковое изображение со своего сервера. Именно поэтому вам понадобится как минимум широкополосное подключение к Интернету: только тогда вы насладитесь в полной мере.

> «Навигация сводится к набору названия места, куда вы хотите попасть.»

Уровень детализации зависит от местонахождения. США, например, могут похвастать наилучшим покрытием: здесь не диво разглядеть автомобиль или даже людей. Детализация других частей света различна, чаще всего она доходит примерно до 15 метров на пиксель.



» Пусть Большой Каньон находится на другой стороне планеты – исследовать фотогеничную американскую бездну можно, не выходя из-за компьютера.

Полёт навигатора

Секстант не понадобится: навигация сводится к набору на клавиатуре названия места, куда вы хотели бы попасть. Вид плавно удаляется в космос, затем перемещается к месту назначения. Приоритеты расставлены с американских позиций – например, если набрать «Birmingham», то вы угодите в Алабаму, а не в более крупный город в Британии.

Прибыв на место назначения, можно щёлкать и перемещать карту мышью (совсем как в Google Maps). Но если нажать и удерживать среднюю кнопку мыши, вы измените угол зрения со стандартного (сверху вниз, как со спутника) до совершенно горизонтального, если захотите. Google Maps так не умеет – а что особенно поражает, *Google Earth* использует высотные данные для приближённого отображения неровностей земной поверхности. Например, посетив Эверест или Большой Каньон, вы увидите их реалистичные модели.

Надоело выискивать собственное местонахождение? Интернет полон открытиями других людей. Они пользуются KML, Keyhole Markup Language, и отмечают интересные места, новые 3D-модели (с текстурами!), фото, клипы и информацию, напрямую связанные с Интернетом. Поиски могут вывести и на рекламодателей Google, аналогично страницам поиска в поисковой машине. Это может пригодиться, если вы, скажем, ищете ближайший ресторан в Непале, а программа в этом помочь не может. Закладки делаются на всех интересных местах, совершенно так же, как в обычном браузере.

Захлопает бабочка крылышками на другой стороне планеты – и *Google Earth* создаст у вас иллюзию, что двойным щелчком мыши можно вмешаться в это событие и наблюдать наступивший хаос. *Google Earth* будит желание поохотиться за метеоритными кратерами или найти исток Амазонки. Бесплатный ресурс такой мощи просто не за что критиковать. LXF

Свойства навскидку



Высота гор
Использование высотных данных означает, что горы типа Эвереста поднимаются над земной поверхностью.



Отображение зданий
В нескольких городах США отображаются даже дома, а пользователи добавили текстурные здания и в другие места.



Вердикт

Google Earth

Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» *Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная программа.*

Рейтинг 9/10

AC3D 6



Linux не обижен инструментами 3D-моделирования, но почти все они – не для обычного пользователя. **Ник Вейч** отыскал среди них не требующий семи пядей во лбу.

Вкратце...

» Платный инструмент объемного моделирования со славной историей. Аналоги - Blender, Maya и Realsoft 3D.

Производители спецэффектов давно влюблены в платформу Linux – она экономит время и деньги в сравнении с другими, менее стабильными, системами. Но их фирменные приложения вряд ли доступны для эффективного использования непрофессионалами.

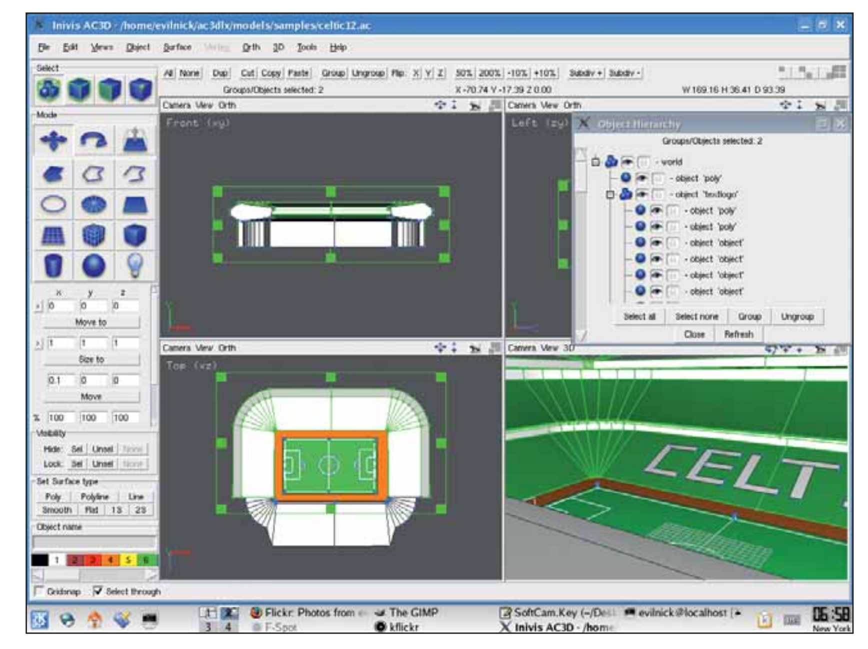
Поприветствуем же *AC3D*, программу, достаточно простую для домашнего пользователя и, при этом, достаточно мощную по всем параметрам. Ныне она празднует свое десятилетие (первоначально *AC3D* создавалась для Unix и Windows), причем все это время она активно развивалась.

Всеобщая совместимость

AC3D гораздо лучше других Linux-инструментов справляется с открытием файлов 3D Studio Max, LightWave 3D, DXF (AutoCAD) и… всех остальных. Огромное количество сред разработки, в которых теперь создаются 3D-объекты, а также проприетарная природа систем рендеринга привели к появлению множества разных форматов. Способность *AC3D* понимать их – одно из тех качеств программы, которые окупают расходы; вот только иногда в процессе трансляции теряются некоторые текстуры и объекты.

> «Если вы устали бороться с превосходным, но сложным Blender – попробуйте это.»

Благодаря кросс-платформенности *AC3D* можно работать даже с теми форматами, которые на Linux обычно не поддерживаются – например, *MilkShape* (эта программа часто используется для моделирования арены компьютерных игр). Возможность экспорта в эти форматы существенна, если необходимо включить в рабочий процесс другие закрытые программы. Фактически, *AC3D* может запросто обращаться ко многим рендерерам (по умолчанию – POV-Ray), хотя это более полезно для Windows/Mac версии.



› От простейших предметов до сложных игровых миров – AC3D умеет делать все.

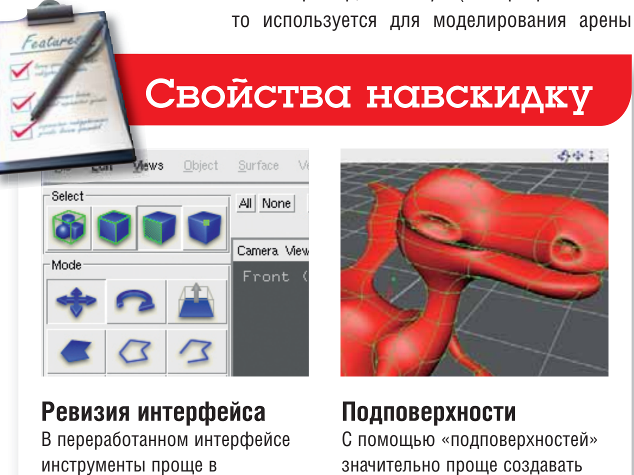
Интерфейс похорошел

В версии 6 капитально обновился пользовательский интерфейс, основные инструменты получили цветные значки, теперь их нетрудно отыскать. По ощущению, программа стала значительно быстрее и отзывчивее, чем в предыдущей версии. Возможно, это просто следствие обновления наших компьютеров с тех пор, как мы в последний раз рассматривали *AC3D* – OpenGL-рендеринг видов тоже заметно порезвел.

Наряду с косметикой, перемены есть и в функциональности. Наиболее заметная из них – оконная иерархия объектов, с помощью которой можно включить/отключить видимость различных объектов или просто убедиться, что редактируется нужная фигура. Новые органы управления ограниченным движением, а также улучшенная видимость границ полей редактирования упрощают редактирование фигур, а изменения в процессе выбора означают, что вам больше не понадобится клавиатурная гимнастика для отметки нужных вершин.

AC3D довольно долго считалась инструментом доводки, ведь в ней отсутствовали многие сложные функции, доступные в других платных программах. Это во многом справедливо: *AC3D* в основном полигональный/вершинный редактор без прихотливых режимов моделирования; но теперь он поддерживает впечатляющую функцию работы с «подповерхностями» (соподчиненными поверхностями), которая существенно упрощает создание «органичных» фигур и плавных изгибов. Если у вас уже есть Maya, то вы, возможно, не захотите связываться с *AC3D* (при всем многообразии поддерживаемых форматов). Если же вы устали бороться с превосходным, но сложным Blender – загрузите (свободно) пробную версию *AC3D*, и попробуйте начать с нее. LXF

Свойства навскидку



Ревизия интерфейса

В переработанном интерфейсе инструменты проще в использовании, да и на вид они проще!

Подповерхности

С помощью «подповерхностей» значительно проще создавать «живые» фигуры.

LINUX FORMAT Вердикт

AC3D 6

Разработчик: Inivis
Сайт: www.ac3d.org
Цена: $69.95

Функциональность	9/10
Производительность	9/10
Простота использования	8/10
Цена	8/10

» *$69,95 – сущая безделица за такой основательный, такой простой и такой полезный товар.*

Рейтинг 9/10

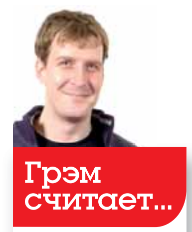
Opera 9



Песенка коммерческой *Opera* уже спета, но шоу, тем не менее, продолжается. **Ричард Коббетт** разбирается в деталях.

Вкратце...

» Новейшая версия комплекта Opera «все-в-одном», с виджетами и поддержкой BitTorrent, по-прежнему отличается высокой скоростью.



Грэм считает...

«Все мы любим Opera, но она предлагает недостаточно, чтобы превзойти способности Firefox или живую многоплановость Konqueror».

Вы жалели *Opera*. Не потому, что он остался в стороне от титанической схватки между *Explorer* и *Firefox*, и даже не потому, что это достойный претендент, так и не дождавшийся своего часа. А потому, что нет другого браузера с такой богатой историей первопроходца в самых крутых вещах – от жестов мышью до вкладок иээ... *Greasemonkey* (хотя в *Opera* это уныло называлось *User JS*, потому-то он и не взлетел) – и тем не менее его вспоминают всего лишь как «браузер с баннерами».

Все, хватит. Теперь *Opera* 9 бесплатен, освобожден от прежних рекламных и лицензионных ключей, готов к действию. И вот вам факт – «на выходе из коробки» это самый вылизанный и функциональный браузер из имеющихся. Пользоваться *Opera* не только просто, но и приятно. До всего, что вам нужно, дотянетесь мышью, и нет нужды ни в каких расширениях. Это полный web-пакет, справляющийся и с простейшим Интернет-серфингом, и с почтой (IMAP и POP3).

Звучит неплохо? Да. Но есть и недостатки. Если сайт построен в духе любимых Дэвидом Картрайтом [David Cartwright] web-стандартов, то проблем не возникнет – *Opera* 9 проходит тест Acid2 и превосходно обрабатывает большинство статических страниц. Но вот многие сайты на Ajax, созданные и тестированные эксклюзивно под *IE* и *Firefox* – другое дело. Google Calendar более-менее совместим, но все еще отмечает *Opera* как неподдерживаемый браузер, а динамические ссылки в инструменте online-заметок Backpack ломаются. Может быть, посещаемых именно вами сайтов проблемы и не касаются – но некоторые страницы просто не работают.

Web-разработчикам, конечно, жить стало легче: встроенные таблицы стилей позволяют посмотреть, как страничка будет выглядеть на экране мобильного устройства, без таблицы стилей или в «сыром» текстовом формате. Можно провести валидацию одним нажатием клавиши, а если вы подготовили подходящую презентацию, используйте *Opera* как инструмент для слайд-шоу.



» **Все, что нужно от браузера, если не вообще все, что нужно – добавка новой системы виджетов к прежде неизменной линейке *Opera*.**

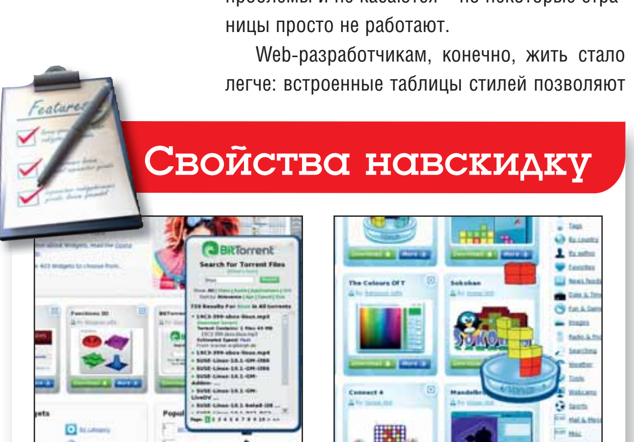
Быстрее всех

В скорости *Opera* равных нет, если говорить о рендеринге страниц и времясберегающих функциях движка. Потребление памяти в нашем тесте составило 28 МБ – меньше, чем *Firefox* и *Epiphany* (оба потянули на 32 МБ). Усиленно рекламируемой на сайте *Opera* голосовой системы управления пока нет, но легкость доступа ко всем инструментам, и множество клавиатурных комбинаций делают эту нехватку несущественной. Один из самых полезных инструментов – диалоговое окно Site Preferences (Параметры сайтов), в котором можно отключить навязчивые cookies и встроенные фреймы или настроить обработку «правого» щелчка мыши, а также добавить собственную таблицу стилей или файл JavaScript, чтобы изменить поведение страницы.

Наиболее значительное дополнение – система виджетов (Widget). При работе *Opera* виджеты порхают друг над другом и над рабочим столом, и установка любого из них – дело одного щелчка. Не удивительно, что поисковый инструмент *BitTorrent* расположен выше всех; сразу за ним следуют напоминания и простые игры – их несложно создать, и они кросс-платформенны.

У *Opera* остались проблемы с просмотром web-сайтов, и вариант «все-в-одном» не похож на *Firefox* с миллионом расширений. Но это совершенно иной стиль браузера, при котором все инструменты работают совместно, а не явились со всех концов сети, и тут есть свои преимущества. Виджеты – превосходное дополнение, а солидный самостоятельный браузер позволяет немедленно окунуться в Сеть с головой. LXF

Свойства навскидку



BitTorrent

Загрузка через встроенный BitTorrent стремительна, как падение с дерева.

Виджеты

Виджеты кросс-платформенные, хотя на Linux наблюдаются некоторые графические проблемы.

Вердикт



Opera 9

Developer: Opera Software
Web: www.opera.com
Price: Бесплатно по закрытой лицензии

Функциональность	9/10
Производительность	9/10
Простота использования	9/10
Документация	7/10

» *Хорош и для продвинутых пользователей, и для простых любителей Интернета, но возможны проблемы с сайтами, использующими Ajax.*

Рейтинг 8/10

Xfce 4.4

Xfce бросает вызов соперникам от Gnome и KDE – но не пришлось ли пожертвовать функциями ради скорости? Разбирается **Майк Сондерс**.

Вкратце...

» Настольная система на базе GTK для X11R6. Аналоги: KDE, Gnome и ряд отдельных оконных менеджеров.

Скорость любят все – мы постоянно ищем способы ускорения системы и стремимся выжать из нашего компьютера все до капли. Выбор ПО – ключевой момент в этом стремлении, и если вы разыскиваете легкий и быстрый вариант, Gnome и KDE – не для вас. Конечно, они нашпигованы функциями, утилитами и библиотеками, но это же настоящие пожиратели RAM! Оливье Фордан [Olivier Fourdan] со товарищи создали более эффективную среду *Xfce*, которая призвана «загружать и исполнять приложения быстро, экономя при этом системные ресурсы».

Многие дистрибутивы включают *Xfce* в свои репозитарии, но команда *Xfce* поступила умно, создав легкий в установке бинарный пакет: – нет необходимости компилировать что-либо из исходных текстов. Там есть оконный менеджер, панель, файловый менеджер и набор утилит – все окружение рабочего стола. По сравнению с геркулесовыми подвигами по установке Gnome и KDE, запуск *Xfce* гораздо проще и с ходу дает ему фору. А как же хваленая скорость? А вот стартует он в два раза быстрее Gnome и KDE; для новейших компьютеров это, может, и не столь важно, но если у вас машина слабее 1 ГГц, то как не полюбить шуструю загрузку полноценной рабочей среды.

> «Проще в установке, чем Gnome и KDE, и стартует в два раза быстрее.»

Одна из главных задач команды *Xfce* – сделать возможной настройку без ручного редактирования конфигурационных файлов, и для этого проведена похвальная работа. *Xfce* – отнюдь не рай для любителей настроек, как KDE, зато не нужно лазить по «скрытым» настройкам *GConf*, как в Gnome. Диалоговое окно Settings Manager (Менеджера настроек) содержит 14 мини-инструментов для редактирования рамок окон, *GTK*-тем, раскладок клавиатуры и др. Все настройки хранятся в директории **~/.config** в виде XML-файлов, поэтому их нетрудно перенести в другие инсталляции *Xfce*.



› Менеджер настроек *Xfce* позволяет настраивать любую часть рабочего стола – рамки окон, темы *GTK*, клавиатурные раскладки и др. – без обращения к командной строке.

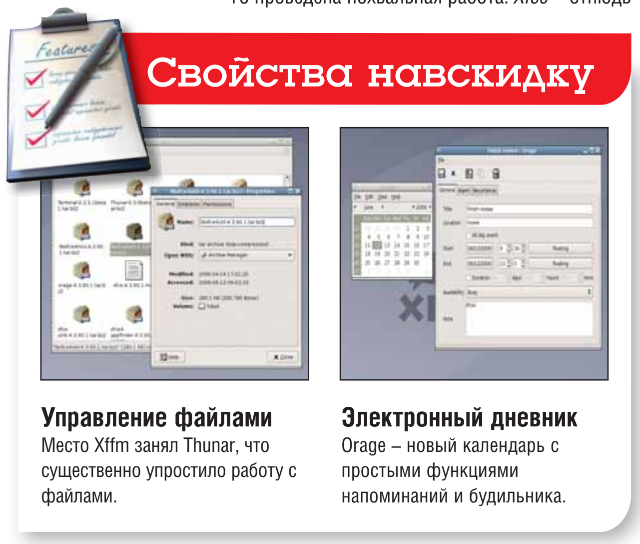
Скандинавское качество

Самое заметное изменение по сравнению с версией 4.2 – включение *Thunar*. Этот файловый менджер пришел на смену работящему, но закосневшему *Xffm*. Thunar намного проще в обращении, слева имеет список часто посещаемых мест, а вверху – кнопки для перемещения взад-вперед по дереву директорий. В духе *Xfce*, он чрезвычайно скор, вдобавок допускает множество изменений в конфигурации, из потерь же по сравнению с *Xffm* ощутима лишь одна – он не может обращаться к сети через Samba. По количеству функций ему далеко до могучих Nautilus и Konqueror, но с основными операциями над файлами Thunar справляется неплохо.

Другие основные усовершенствования включают поддержку иконок на рабочем столе (требуемую пользователями со времен ранних версий 4.x), спартанский текстовый редактор Mousepad и календарь Orage, позволяющий настроить напоминания и придающий *Xfce* оттенок персонального органайзера. Панель обновлена таким образом, что дополнительные функции добавлены в виде модулей, и крушение одного приложения не потянет за собой всю панель.

В нашем тесте *Xfce* бесспорно превзошел Gnome и KDE по части производительности, особенно по использованию памяти панелью и файловым менеджером. Как следствие, с ним можно работать даже на машинах с 64 МБ ОЗУ, хотя и выгрузив все лишнее. Все же компактная кодовая база реже требует сброса на винчестер и обратно при переполнении памяти.

Конечно, *Xfce* нельзя сравнивать с Gnome и KDE по функциональности, но, как нам кажется, соотношение «скорость/возможности» у него совсем неплохое; это привлекательный и крепко сбитый рабочий стол, если вы работаете только с ПО на *GTK*. Если вас достала манера Gnome «прятать» свои функции и настройки, но вы по-прежнему ищете что-нибудь для работы с вашими любимыми *GTK*-приложениями – попробуйте *Xfce*, не пожалеете. LXF



Свойства навскидку

Управление файлами

Место Xffm занял Thunar, что существенно упростило работу с файлами.

Электронный дневник

Orage – новый календарь с простыми функциями напоминаний и будильника.



LINUX FORMAT Вердикт

Xfce 4.4

Разработчик: Команда Xfce
Сайт: www.xfce.org
Цена: Бесплатно по лицензии GPL

Функциональность	7/10
Производительность	9/10
Простота использования	7/10
Документация	8/10

» *В погоне за скоростью Xfce создал прекрасную настольную систему, особенно с введением Thunar.*

Рейтинг 8/10

Gorky 17

Эта игра с польским уклоном посвящена опасным последствиям большевистско-фашистских биологических экспериментов, пишет **Грэм Моррисон**. Приканчим же мутанта!

Вкратце...

» Напоминает о point-and-click боевиках типа замечательной UFO: Enemy Unknown. См. также: UFO2000.

Из названия игры ясно, что действие *Gorky 17* происходит на пост-апокалиптических, пост-коммунистических пустошах Восточной Европы – а это самое подходящее место для нашего отпуска. Добро пожаловать в польский город Любин [Lubin], который, видимо, стал жертвой жестоких исследований в биотехнологии и разработок советского правительства, и улицы его теперь совсем небезопасны. Вот куда вы попали. Вы, офицер Коул Салливен [Cole Sullivan], возглавляете группу из трех солдат НАТО, которую послали в Любин узнать причину отсутствия связи с городом, а заодно и судьбу первой группы натовских солдат, которые выкрикнули по радио свои последние бессвязные слова несколько часов назад.

Пальба как в 1999-м

Анимированное вступление (разрешение низкое, но атмосферу создает) открывает жуткое зрелище: разрушенный город осажден неведомой силой, с ней-то вам и предстоит сразиться. Действие *Gorky 17* происходит в ближайшем будущем, но ее исходная версия для Windows вышла в далеком теперь 1999-м, поэтому не ждите новейших 3D-эффектов, использования пиксельных шейдеров или других модерновых технологий. С тех пор все перешли на графические карты совершенно иного уровня, а из Nvidia и ATI удалось выжать драйверы для Linux. Единственное преимущество игры состоит в чрезвычайно скромных системных требованиях: 300 МГц и 64 MB ОЗУ. Она пойдет даже на PPC!

Движок игры состоит из изометрического фона, с заранее тщательно прорисованными мелкими деталями; основные персонажи нанесены поверх при помощи примитивного 3D-движка. Качество текстур и моделей низкое, но можно переключиться из режима стандартного рендеринга в аппаратно-ускоренный режим OpenGL. Уровень *Unreal Tournament 2007* этим не достигается, зато смягчаются тени и резкие края, которые выдает программный рендерер. Но фон отрисован достаточно детально, чтобы увлечь. Та же техника используется в играх типа *Baldur's Gate* – и этим



› **Готовый фон прорисован чрезвычайно подробно, как видно по этой сцене в музее. Увы, динозавр не оживет.**

> «Город кишит созданиями вроде биомеханоидов Ганса Руди Гигера.»

Шаг за шагом: Поочередная ликвидация монстров



» Место в строю

В режиме схватки пространство вашего перемещения подсвечено зеленым. С каждым вашим движением это пространство постепенно сужается, пока не истечет время вашего хода.



» Выбор оружия

Необходимо выбрать место, с которого ваше оружие сможет достать плохого парня. Не помешает позаботиться и о безопасном отступлении.



» Убить гада

Когда вы увидите белки его глаз, прицельтесь и всадите ему заряд сами-знаете-куда. Осторожно! Мутанту могут придти на помощь отпрыски-клоны.

сходство не ограничивается. Как и в *Baldur's Gate*, есть два игровых режима: стратегия и схватка.

Битва с мутантами

Первый режим – просто разведка на местности. Покружите курсором по экрану, щелкните в какой-нибудь точке – и ваша доблестная команда направится туда, обходя препятствия на пути. Если это место недосягаемо, из курсора вылетит маленький красный кружок, а команда останется стоять.

Основная часть игры проходит во втором режиме. Когда ваши герои доберутся до определенной точки, игра перейдет в режим схватки. Фон игры изменится на предварительно отрисованный, с полом, разлинованным на квадраты, и сюжет переключится с point-and-click реального времени на стратегию поочередных ходов. Именно здесь у *Gorky 17* есть шанс вам понравиться, так как режим схватки очень похож на всеми нами любимый UFO: *Enemy Unknown*. Каждый персонаж за один ход может передвинуться на определенное количество квадратов и воспользоваться одним объектом. Чаще всего этим объектом бывает оружие, но может быть и лекарство или бинт.

Стратегия исходит из сравнения возможностей оружия у вас и ваших противников, ибо возможности эти ограничены. Например, пистолеты подобны ладьям в шахматах и могут прицеливаться только по вертикали и горизонтали. Винтовка уже напоминает ферзя и может стрелять также по диагоналям. По ходу битвы вооружение набирает мощь, и к моменту решающей схватки мы обладаем уже метателем жидкого азота и огнеметом. В шахматах это равносильно перевороту доски.



› В игре есть закадровые диалоги, но голоса персонажей несколько раздражают.



Пол считает...

«Эта игра для *UFO* – то же, чем *X2* была для Elite: неплохой клон, но лишенный той магии, что делала притягательным оригинал. Лучше скачайте с eBay копию *UFO* и опробуйте *DOSBox*».

В *Gorky 17* есть и элементы ролевой игры. По ходу действия каждый персонаж приобретает опыт, повышает свой уровень и набирает статистику. Но это не та статистика, которая обычно ведется в ролевых играх. Правда, статистика для очков имеется, но все остальные связаны с многоходовой битвой. Например, есть счет, управляющий количеством квадратов, которые можно пройти за один ход, а есть подсчет точности ваших выстрелов. Есть также учет Удачи и Спокойствия, связанный со снятием нескольких очков у вашего противника в случае удачи и сохранением вашего спокойствия, когда дела идут плохо.

А схватки жестоки. Первый сценарий, по сравнению со всей остальной игрой самый лобовой – морской порт города. Перед боем наши бесстрашные разведчики натыкаются на двоих павших товарищей. Хитрость состоит в том, чтобы выманить противника в пределы дальности оружия и успеть убраться восвояси до ответного хода врага. Врага? Есть враг: город наводнен созданиями, похожими на биомеханоидов Ганса Руди Гигера. В отличие от случайных персонажей, эти отрисованы игровыми дизайнерами, и крупные сражения включают клипы-врезки, знакомящие с каждым из фотогеничных «героев».

Heaven 17?

Итак, что же за игра *Gorky 17*? Любая местность выглядит прекрасно, упавшие статуи в музеях и мерцающий неон над городскими магазинами реально захватывают. Сюжет тоже хорош, но шансов насладиться им мало. Скорее всего, 95% времени уйдет на битвы, а остальное – на поиски ключа или дерганье рубильника. Продолжительность игры зависит от того, как много схваток вы сможете выдержать. Сражения не достигают того уровня интенсивности и накала, к каким мы привыкли в UFO – они обычно сводятся к подсчету затраченной вами амуниции или числа канистр с горючим, находящихся поблизости от мутантов (канистры можно подорвать с помощью выстрелов, чтобы нанести максимальный ущерб). Это и не вполне стратегия, скорее легкое развлечение.

Тот факт, что игра реанимирована из 1999 года, пробудил было в нас надежду, что это забытая классика. Вышло не совсем так, но и полного забвения она отнюдь не заслуживает. LXF



› К счастью, плохие парни оставили уйму амуниции и прочего добра – только успевай подбирать.



Gorky 17

Разработчик: Linux Game Publishing
Сайт: www.linuxgamepublishing.com
Цена: £25.00 (ок. $40)

Сюжет	7/10
Графика	6/10
Долговечность	5/10
Цена	5/10

» Дороговато для игры, которую можно найти на барахолке за гроши, но нищие не выбирают!

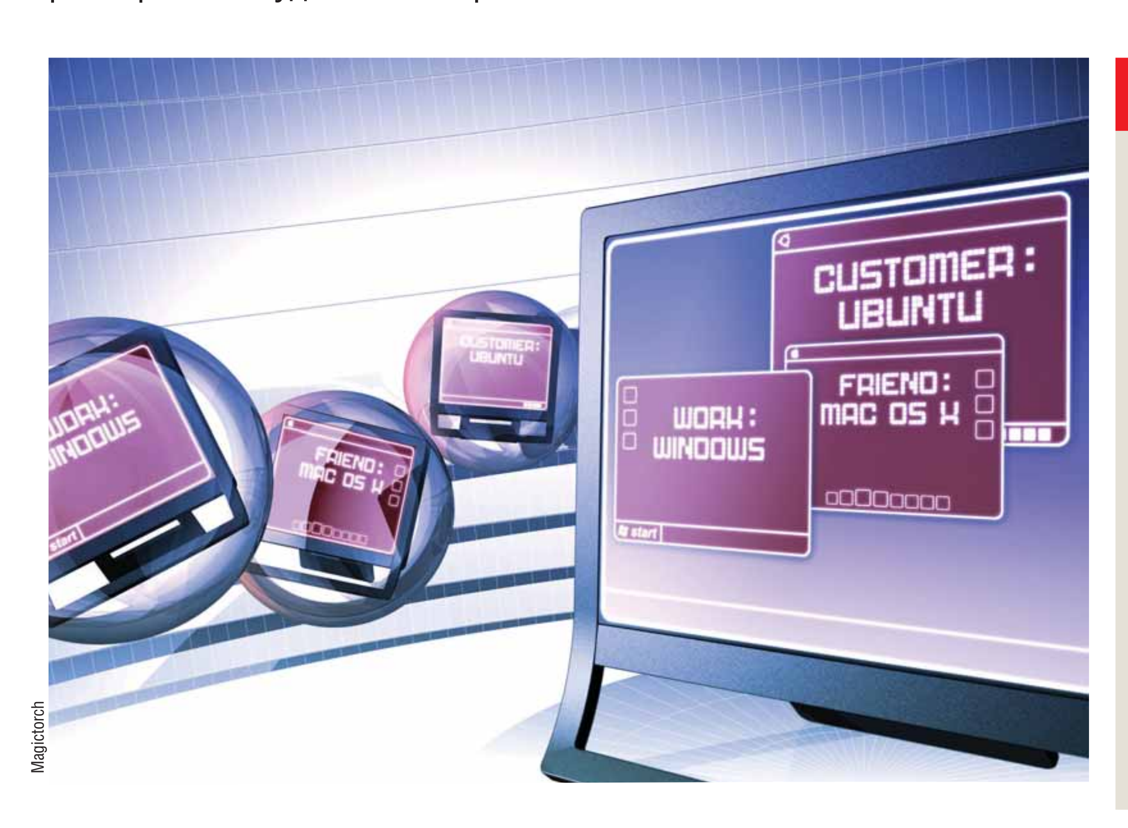
Рейтинг 6/10

Сравнение

» Каждый месяц мы сравниваем для вас тонны программ — а вы можете отдыхать!

Утилиты удаленного управления

В этом месяце: Чтобы не держать на столе лишние мониторы, **Дэвид Кулсон** разбирается с удаленными рабочими столами.



Magictorch

Наши тесты

Удаленный рабочий стол должен обеспечивать пользователя всеми преимуществами рабочего стола (а не консоли) на удаленной машине. Ключевые факторы здесь таковы:

» **Скорость**. Рабочий стол с непомерно медленной реакцией бесполезен.

» **Безопасность**. Не рисково ли использовать его в незащищенной сети? Можно ли зашифровать данные и потребовать аутентификации?

» **Внешний вид**. Хорошо ли воссоздается графический рабочий стол, возможно ли сжатие передаваемой информации?

Мы использовали две тестовых конфигурации. Первый тест был через локальную сеть 100 Мбит/сек, между двумя Linux-системами . Этот тест показал нам, на что способны протоколы при условии стабильной связи. Второй – через кабельное соединение 3 Мбит/сек с удаленной сетью, при задержке около 40 миллисекунд. Это типичные параметры сети для работы и из дома, и через WAN между офисами.

Наш выбор

Помогла бы вам иной раз возможность увидеть рабочий стол Windows вашего клиента на своей Linux-машине? Приходилось ли вам готовить доклад на понедельник в выходные? Или вы просто любите работать на дому, развалясь в пижаме? Вам, друзья мои, нужен удаленный рабочий стол.

Доступ к рабочему столу сервера или рабочей станции – функция, уже десятилетиями предлагаемая на платформах Unix, но до недавнего времени это был просто терминальный доступ, как у большинства пользователей. К счастью, в наши дни существуют разные варианты графического доступа. Пусть вы их еще не пробовали – вы тоже можете увидеть свой рабочий стол так, словно он установлен в другом месте (и даже на другой ОС), и открывать графические приложения. Движения вашей мыши и нажатия на клавиши будут отсылаться на сервер; в ответ обновления экрана отправятся к вашей удаленной машине. Есть несколько опций для кодирования и сжатия этой экранной информации, чтобы «перерисовка» всего графического рабочего стола точка за точкой не отнимала половину вашего канала и процессорного времени.

X11 и далее!

Рабочие столы Linux основаны на протоколе X11, со встроенными сетевыми возможностями, позволяющими удаленным системам запускать графические X-приложения на локальном X-сервере. Это имеет существенные недостатки: главным образом то, что вы не можете запустить удаленно X-сессию целиком и что это «отъедает» пропускную способность. Даже запустив X11 через Ethernet 100Mbit, вы можете ощущать задержку, способную кое-кого вывести из терпения. Мы рассмотрим X11 в этом Сравнении, но есть и альтернативы запуску «сырого» *X* по сети, которые мы тоже протестируем; многие из них включают дополнительные функции, далеко превосходящие возможности X11.

Всего мы рассмотрели семь решений удаленных рабочих столов: три полных клиент-серверных пакета, два сервера и два интерфейса (front-end), которые вам следует комбинировать, чтобы получить работающее решение удаленного рабочего стола.

Клиент-сервер



VNC Free Edition

Оригинальный мультиплатформенный протокол удаленного управления.

В середине 90-х сотрудники лаборатории AT&T в Кембридже соорудили довольно изящный протокол, Virtual Network Computing (VNC), спроектировав его для удаленного контроля, позволяющего любой платформе управлять любой другой. Серверы и клиенты VNC были разработаны для Unix, Windows и Mac OS, наряду с набором мобильных устройств. Многие из этих реализаций были почти бесполезны, но они демонстрировали гибкость VNC, обязанную использованию собственного внутреннего протокола – Remote FrameBuffer, или RFB. Спустя десять лет после выхода VNC, RFB продолжает быть основой почти всех систем удаленных рабочих столов в Linux.

Открытая реализация VNC, включающая сервер и клиентский просмотрщик (viewer) для удаленного доступа к данным, использовалась коммерческими компаниями и домашними пользователями с 1998 года. Но RealVNC, фирма, основанная выходцами из AT&T Laboratory, выпустила также три коммерческих VNC-решения, включая версии для предприятий и персонального использования. В нашем Сравнении мы тестировали свободную версию, известную также как RealVNC, используя ее с рабочим столом Gnome и клиентами Linux и Windows.

Технология довольно проста. Сервер VNC запускается на машине Linux, предоставляя новую X-сессию, через которую запускаются приложения. Типичная X-сессия на рабочем столе Linux – :0.0, так что *VNC* следует искать запущенным на :1.0. Однако, в отличие от стандартного *X*-сервера, *X*-сервер *VNC* (*Xvnc*) не полагается на доступность физического видеоустройства, а запускается независимо от любого оборудования на локальной машине.



› **Множество применений *VNC* выходят далеко за рамки этого простого примера: «X в X», показывающий Mac и Linux на машине Windows.**

Чтобы поработать с *Xvnc*, вы подключаетесь к нему удаленно через RFB, используя программу-клиента VNC (просмотрщик) – и готовая X-сессия открывается в окне.

Те, кто «подсел» на терминал, знакомы с программой screen – она позволяет запущенным удаленно (через SSH или Telnet) процессам выполнять свою работу и осуществлять вывод, даже если сессия была завершена. Позже пользователь позже может снова подключиться к машине, запустить screen и продолжать работу, как ни в чем ни бывало. *VNC* действует очень похожим способом: даже если ваша сеть «упадет», сессия *Xvnc* не прекратит работу, и все X-приложения – тоже. Это удобно: пользователь может отсоединиться от сессии, чтобы перезагрузить свою рабочую станцию или подключиться из другого места.

Кодирование

VNC предлагает ряд методов кодирования, предоставляющих множество способов перерисовки рабочего стола: от наиболее общей реализации «raw», которая обновляет буквально каждую точку, как только она изменится, до сложных методов hextile и corre, допускающих меньшую пропускную способность и перерисовывающих изменения быстрее. Впрочем, ни один из методов кодирования на самом деле ни сжимает данные, ни модифицирует их методом с потерями, так что VNC на медленном соединении будет слегка срываться (см. врезку слева «Стоит ли ждать»). Однако на Ethernet 100Mbit VNC почти не отстает от обычной X-сессии – мы даже смогли через сеть проиграть DVD на Totem, без заметного дрожания или рассинхронизации звука и изображения.

Одна из довольно полезных возможностей VNC – слежение (shadowing) за рабочим столом, когда несколько пользователей могут подсоединяться к одному серверу *Xvnc* и либо разделить сессию, либо передать управление одному из них, а остальных – сделать наблюдателями. Это приложение идеально для тренингов или конференций, когда несколько человек работают над проектом вместе. Каждый, кому приходилось искать неисправности и кто хочет, чтобы поставщик или консультант понаблюдали за проблемным пользователем, найдет его бесценным.

Безопасность RFB

Свободная версия VNC не предусматривает шифрования, так что запуск сессии RFB через открытое интернет-соединение сравним с использованием Telnet. Пароли для VNC-аутентификации и нажатия клавиш во время сессии передаются открытым текстом, и их легко перехватить. Однако в Linux уже есть многочисленные методы обеспечить безопасность IP-трафика, либо используя туннель VPN, либо туннелируя RFB-сессию через шифруемый порт SSH. С учетом доступности утилит шифрования и простоты их развертывания, все, кто планирует использовать VNC для работы, просто придут на готовенькое.

Если требуется удаленный X-доступ из Linux, использования в той или иной форме VNC практически не избежать. Но данная версия лишена функций сжатия и аутентификации, и пользователям, желающим чего-то большего, чем базовое решение, лучше присмотреться к *TightVNC*. Если какие-то из устройств его не поддерживают, всегда можно использовать в роли клиента обычный VNC и, благодаря обратной совместимости с RFB, иметь возможность подключаться к системам *TightVNC*.

Стоит ли ждать?

Те из нас, кто пользуется VNC несколько лет, в курсе, что он очень удобен и гибок на соединениях с высокой скоростью и низкой латентностью, но становится практически бесполезен, как только скорость передачи данных, требуемая для обновления экрана, превышает возможности локального соединения. Типичная ситуация, когда это становится проблемой – удаленный офис с ADSL- или кабельным соединением, где скорость исходящего потока заметно ниже, чем входящего. Пропихивание целой X-сессии через 256 кбит/сек не слишком радует, даже при обычном запуске Twm или Xterm.

> «В Ethernet на 100 Мбит/сек VNC почти не отстает от X-сессии.»

LINUX FORMAT Вердикт

VNC Free Edition

Версия: 4.1
Сайт: www.realvnc.com
Цена: бесплатно по лицензии GPL

» *Дедушка всех приложений удаленного рабочего стола в Unix рискует быть потесненным внуками, рассмотренными в нашем Сравнении.*

Рейтинг 6/10

TightVNC





Поднимает VNC на новые высоты.



› При JPEG-сжатии медленное соединение может передавать сложные рабочие столы, но качество изображений становится убогим.

TightVNC – это VNC, сделанный лучше, быстрее... компактнее [tight – компактный, сжатый, – прим. пер.]. Созданный командной российских разработчиков как улучшенная версия VNC, *TightVNC* предлагает альтернативные расширения для протокола RFB, предоставляющие улучшенную технологию сжатия для пользователей с медленными соединениями. Это достигается ценой небольшого увеличения латентности, пока рассчитывается сжатый рабочий стол, а также увеличения нагрузки на процессор, но это не повод для беспокойства для процессоров, созданных за последние лет пять.

Как и *RealVNC*, мы тестировали *TightVNC* с рабочим столом Gnome и клиентами Linux и Windows. Методы сжатия могут быть либо «без потерь», либо JPEG. Варианты «без потерь» включают «fast» (быстрое), для наиболее отзывчивого рабочего стола, и «best» (наилучшее). Хотя опция JPEG (сжатие с потерями) может попортить ясность и четкость рабочего стола, она реально снижает требования к пропускной способности. Результирующее качество варьируется от «почти оригинал» до «клякса какая-то». JPEG-сжатие с низким качеством определенно полезно, но оно не для тех, кто любит мелкие шрифты или заинтересован в детальном изображении. При низком качестве удаленного стола шрифты, изображения, меню и т.д. фактически нечитаемы, потому что все они растровые, и просмотр интернет-страниц почти невозможен. Зато JPEG-сжатие работает через модем 56 кбит/сек – всяко лучше, чем ничего.

В *TightVNC* отсутствует «родное» шифрование, но имеется ряд скриптов, которые позволяют ему работать через stunnel (оболочка к SSL-шифрованию), предоставляя TCP-сессию между клиентом и сервером, зашифрованную с помощью SSL.

Нужен быстрый удаленный доступ через медленное соединение? *TightVNC* как раз для вас. Нужен быстрый локальный доступ через LAN? Все равно берите *TightVNC*, потому что все функции обычного VNC у него есть. Короче говоря, пусть все пользуются *TightVNC*.

> «Предлагает улучшенную технику сжатия для медленных соединений.»



TightVNC

Версия: 1.2.9
Сайт: www.tightvnc.com
Цена: бесплатно по лицензии GPL

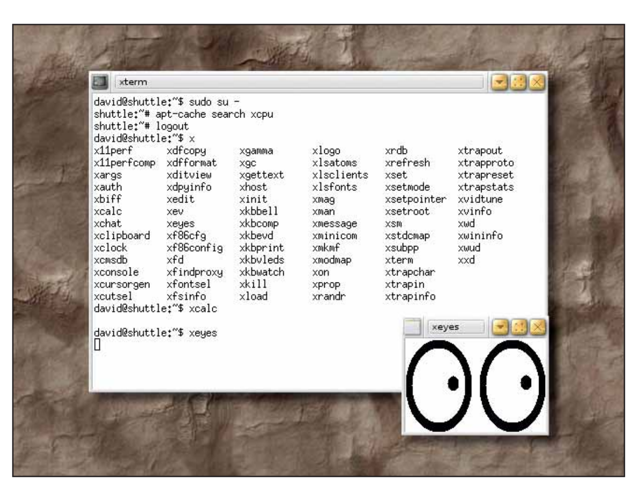
» *С дополнительными функциями сжатия, TightVNC – лидер нового поколения реализаций VNC.*

Рейтинг 9/10

X11



Linux-сервер, используемый удаленно.



› Это приложение запущено через сеть. Не похоже? Поверьте, так и есть.

Все, что вы делаете с X, работает через протокол, основанный на сокетах, либо TCP, либо Unix. Мы вспомнили об этом в Сравнении удаленных рабочих столов, потому что X позволяет приложениям подключаться через сеть и к локальной X-сессии, и к другой системе. Все возможности X-сервера будут доступны клиенту: например, если удаленная машина не поддерживает GLX (как же мы без OpenGL?), вы все равно сможете запускать GLX-приложения, хотя производительность слегка пострадает. X – это сервер, но клиентом является приложение: вы ничем не подсоединитесь к нему. Мы тестировали X.Org 7.1.

Все *X-приложения* полагаются на переменную **DISPLAY**; изменив ее, чтобы она указывала на другую систему, вы можете прозрачно запускать приложения (но не весь рабочий стол) на другой машине, как будто они работают локально. Ликуйте все, кто работает на нескольких машинах и кому уже некуда ставить дополнительные мониторы! Один из основных недостатков X11 – очень простой, основанный на событиях протокол, и вы вряд ли захотите запускать его через общедоступную сеть. Поскольку все работает с использованием TCP, «обернуть» сессию в SSH или stunnel довольно легко; и как только туннель будет построен, можно будет исполнять различные приложения с использованием шифрованного соединения. Но и тогда вы должны убедиться, что сессия SSH работает постоянно, не то по завершении сессии все X-приложения бесславно погибнут.

X11 через сеть может пригодиться, но вообще-то он проектировался для использования исключительно в среде LAN. Он умеет работать через VPN, но опасность потерять все ваши *X*-приложения из-за кратковременного сбоя сети, в общем, слишком велика, чтобы он стал действительно полезным.

> «Можете прозрачно запускать приложения на другой машине.»



X11

Версия: 7.1
Сайт: www.x.org
Цена: бесплатно по ряду открытых лицензий

» *Превосходен для локальной сети, но для прочего хрупковат, хотя можно попробовать его с туннельным соединением или VPN.*

Рейтинг 4/10

X11vnc



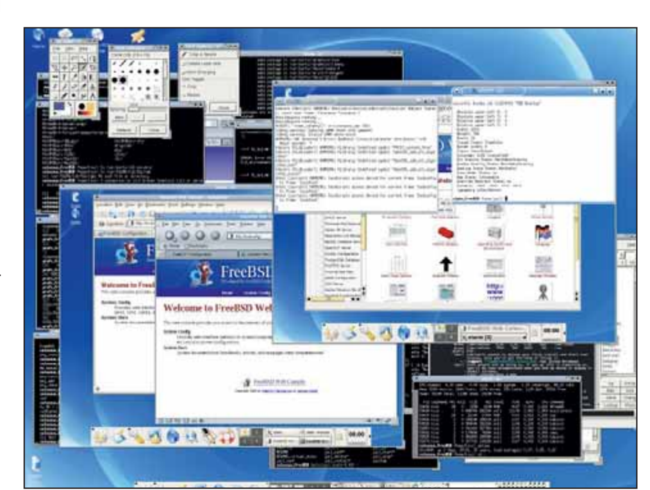


Используется с VNC-просмотрщиком для экспорта сессий X11.

Написанный в 2002 году, *X11vnc* запускается как клиент к любой сессии X11, которую вы хотите экспортировать для удаленного доступа, предоставляя рабочий стол как VNC-сервер по сети. Поскольку это VNC-сервер, он использует протокол RFB, а не X11. Когда *X11vnc* запущен (мы испытывали его с X.Org 7.11 и клиентом *TightVNC*), удаленный пользователь может подключиться к рабочему столу, либо разделяя сессию, либо только в режиме просмотра. Для просмотра рабочего стола можно использовать различное кодирование, допускающее сжатие и изменение качества JPEG при отображении на стороне клиента, так что даже если у вас на рабочем столе творится нечто безумное, все равно можно будет поглядеть на это через VNC.

Звучит хорошо; но затем возникает вопрос: «Умеет ли *X11vnc* делать то-то и то-то?». Беглый взгляд на **X11vnc -h** даст ответ, правда? После просмотра 40 страниц опций, ключей и других аргументов, вы, может быть, найдете то, что искали, а может, и нет. *X11vnc* – из тех приложений, куда люди приходят и говорят: «Мне нужно это» – и «это» добавляется, будь оно хоть рукомойником или другим столь же нетипичным приспособлением. Многие дополнения и разумны, и полезны, но когда нужен всего-навсего X-дисплей, в поисках помощи можно зарыться в горах информации и навеки. Впрочем, беда невелика: подробное ЧАВО (FAQ) на сайте прекрасно посодействует заставить общую X-сессию вести себя как надо.

Если у вас есть X11 и вам нужно издалека добираться до своего рабочего стола, берите *X11vnc* и не ищите ничего другого – разве что захотите воспользоваться инструментами, встроенными в Gnome или KDE; но даже они и близко не подходят к некоторым из возможностей *X11vnc*. Если вы уяснили все опции командной строки и оптимально настроили конфигурацию, *X11vnc* станет лучшим средством экономии времени и продуктивным инструментом, необходимым каждому рабочему столу X11.



› Экспорт рабочего стола *X11* при помощи *X11vnc* – запросто: можно делать почти все, информацию бы только потолковее.



X11vnc
Версия: 0.8.2
Сайт: www.karlrunge.com/x11vnc
Цена: бесплатно по лицензии GPL

» *Должен быть у каждого, кто использует X как рабочий стол и часто нуждается в удаленном доступе к своей системе.*

Рейтинг 9/10

NX



Новаторская, быстрая, но проприетарная альтернатива VNC.

RFB – прекрасный протокол для базовых однопользовательских соединений. Но ему недостает реальной системы аутентификации, и в более крупных системах проверка машин и слежение за подключениями превращается в кошмар. *NX* не использует ни RFB, ни протокол RDP (Remote Desktop Protocol, протокол удаленного рабочего стола, широко применяемый в Windows) – у него собственный протокол для туннелирования сжатого X и различных протоколов через зашифрованный канал.

Четыре сервера уровня предприятия доступны по отдельности и за плату, а открытый настольный сервер можно скачать с www.nomachine.com, но вам придется использовать его с клиентом *NX* и узлом, бесплатными, но не свободными. Мы рассмотрели один из пакетов уровня предприятия *NX* 2.0.0, протестировав его с X.Org 7.1 и Windows XP Pro.

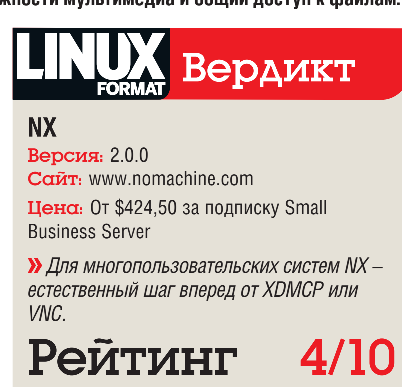
NX необычен тем, что дает возможность удаленного доступа к системе и позволяет приложениям X-клиентов получать доступ к удаленному X-серверу. Это решает проблему отсутствия хорошего способа шифрования или сжатия данных между X-сервером и клиентом: *NX* запускает X Agent, в итоге получается конфигурация в стиле «X внутри X», где один X-сервер посылает обновления другому как события вызова от клиента.

Имея возможность пересылать звук и файлы через одно и то же *NX*-соединение, *NX* превосходен для всех, кто использует удаленный доступ в ежедневной работе. Хотите проиграть Flash-ролик на удаленной машине? Аудио автоматически передается через шифрованное соединение на локальной машине, обеспечивая полное мультимедиа-взаимодействие с вашим удаленным рабочим столом. Как и предоставление файлов в общий доступ, это много лет входило в RDP на Windows, но *NX* приносит это на открытый рабочий стол, плавно интегрируя существующие хосты Windows и новые Linux-инсталляции.

> «NX плавно интегрирует хосты Windows и Linux-инсталляции.»



› *NX* – очень гибкая система, предоставляющая функции *VNC*, дополнительные возможности мультимедиа и общий доступ к файлам.



NX
Версия: 2.0.0
Сайт: www.nomachine.com
Цена: От $424,50 за подписку Small Business Server

» *Для многопользовательских систем NX – естественный шаг вперед от XDMCP или VNC.*

Рейтинг 4/10



Tsclient



Многосторонний инструмент Gnome для протоколов RDP и RFB.

Управление системами, к которым мы получаем удаленный доступ, довольно нудная штука: нужно помнить все имена хостов или IP-адреса и все пароли, используемые для обеспечения безопасности. Для решения этой проблемы можете попробовать *Tsclient* – приложение Gnome, предоставляющее интерфейс ко всем обычным клиентам удаленных рабочих столов, включая *VNC Viewer* (интерфейс к *RealVNC* или *TightVNC*) и *Rdesktop*, что позволяет получать удаленный доступ и к Linux-, и к Windows-системам, а фактически ко всему, что поддерживает RFB. Для этого Сравнения мы испытывали его с серверами Tight*VNC* и RDP.

Tsclient изрядно примитивен. Конфигурация каждой хост-системы сохраняется в файле, загружаемом через интерфейс. Хотелось бы, конечно, небольшого меню, включающего все хосты, а то и организующей их древовидной структуры; зато можно сохранять сессии как для хостов RFB, так и RDP, и учесть все ваши потребности к доступу в одном приложении.

Перейдем к недостаткам. Богатство настроек для RDP компенсируется бедностью ассортимента опций *VNC*. Мы не смогли назначить ни свое кодирование, ни степень сжатия, или поместить его в SSL-туннель. Правда, все больше людей использует удаленный доступ с *VNC* и он интегрируется в большее число окружений рабочих столов, так что опции хостов *VNC* явно ждет улучшение.

Главная головная боль с *Tsclient* начинается, когда доходит до сохранения сессий. Пароли он вообще не сохраняет, что не радует, когда автоматическое восстановление соединения терпит неудачу и система должна дожидаться, пока кто-то не выполнит ввод вручную. Если удаленных систем дюжина, на ввод всех паролей уйдет масса времени. В *Tsclient* есть над чем поработать, чтобы набор функций стал сравним с предоставляемым протоколами удаленных рабочих столов.

> «Все больше народу использует VNC, и опции хостов ждет улучшение.»



› *Tsclient* подключается почти ко всем протоколам удаленных рабочих столов. Попробуйте его как интерфейс к *TightVNC*.



Tsclient

Версия: 0.148
Сайт: www.gnomepro.com/tsclient
Цена: бесплатно по лицензии GPL

» *Милый маленький интерфейс пользователя, но не способен установить многие опции, например, уровень сжатия и шифрование.*

Рейтинг 5/10

Krdc



KDE-интерфейс для доступа к хост-системам.

Пусть *Tsclient* фокусируется на пользователях Gnome; поклонникам KDE тоже не дадут пропасть. *Krdc* – это ответ KDE на требования доступа к сессиям RDP и RFB на удаленных машинах, и (говоря не для драчки Gnome vs KDE) он куда лучше, чем *Tsclient*. Все хорошо организовано, есть список сохраненных хостов, доступных по двойному щелчку мыши.

Персональные настройки чисто базовые, но *Krdc* дает выбор быстрого, среднего или медленного типа соединения, в противоположность *Tsclient*, где вообще нет интерфейса для сжатия. В основном, *Krdc* прилично подбирает сжатие для различных скоростей соединения, но его утверждению, что средний уровень достаточен для некоторых DSL- или кабельных соединений, не очень верится. Как всегда, требуется время, чтобы все настроить и добиться хорошей работы. *Tsclient* предоставляет функцию, которой нет в *Krdc* – это поддержка RDP 5, она пригодится, если нужно вырезать и вставлять текст между удаленной Windows-системой и локальным хостом. Мы лишь установили соединение RDP 4 с помощью *Krdc*; иными словами, копировать данные с хоста на хост непросто.

Эх, вот бы нам интегрированный клиент удаленного рабочего стола, позволяющий размещать иконки, или даже простые меню, на рабочих столах удаленных машин! С длинным хвостом хостов, накопленным *Krdc*, пора уже сделать небольшую панель инструментов для быстрого доступа к серверам или рабочим станциям. Определенно, грань между локальным хостом и удаленной системой должна стираться для конечных пользователей, которые не вполне уловили концепцию возможности доступа к их рабочему столу из другого города.

Так или иначе, *Krdc* опережает *Tsclient* по удобству и простоте, хотя обоим нужна доработка, чтобы быть на уровне.

> «Прилично подбирает сжатие для различных скоростей соединения.»



› Krdc, поставляемый с KDE, предлагает простой в использовании интерфейс на все прихоти клиента удаленного рабочего стола.



Krdc

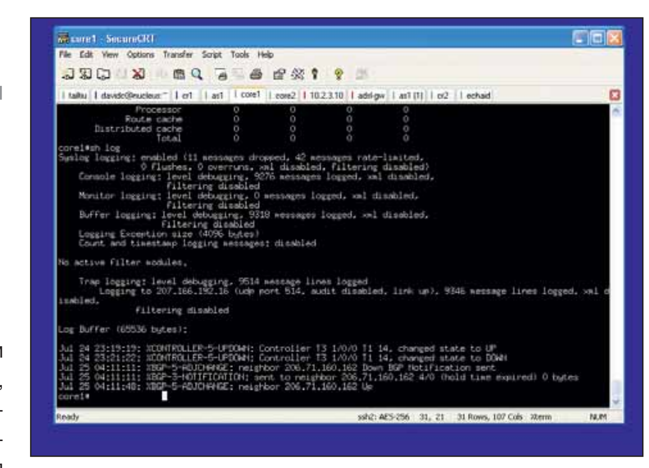
Версия: 3.5.3-1
Сайт: www.kde.org
Цена: бесплатно по лицензии GPL

» *Если вам нужно запускать несколько сессий, Krdc – инструмент для вас; он лучше, чем Tsclient, даже если вы используете Gnome.*

Рейтинг 8/10

Утилиты удаленного управления

ВЕРДИКТ



› *TightVNC*: пусть мы идем по жизни всего с одной парой глаз, это не значит, что они не могут быть многозадачными.

TightVNC 9/10

Поскольку из рассмотренных нами приложений все, кроме двух, ведут свою родословную от *VNC*, понятно, что *TightVNC* – наиболее успешная реализация. При своей гибкости сжатия сессий для удаленного использования и поддержке *VNC*-кодирования без потерь, *TightVNC* удовлетворяет требованиям всех пользователей. Кодирование *TightVNC* поддерживают многие сервера, включая RFB-клиенты и сервера только для Windows, в том числе, *UltraVNC*. Это абсолютный победитель данного Сравнения.

По серверной части *X11VNC* оставляет в кильватере всех. На беду, здесь так много опций, что заставить его работать правильно удастся не сразу. Мы готовы подтвердить его способность упрощать жизнь, поскольку наблюдали, как приложение остается запущенным удаленно, пока мы просто «перепрыгиваем» на машину через SSH и запускаем *X11VNC* для соединения с рабочим столом. Gnome и KDE предлагают функции удаленного рабочего стола на базе *X11VNC*, чтобы интерфейс пользователя стал понятнее и поделился набором опций во всей полноте. Опытным пользователям *X11VNC* под силу соединяться с не-*X11* устройствами, включая фрейм-буферы или даже интерфейсы Video4Linux.

Локальное решение

X11 имеет свое место под солнцем, и так как это оконная система de *facto* на Linux, еще некоторое время будет его сохранять. Простой вызов удаленного приложения через локальную сеть делает *X11* трудным соперником, а ведь ему присуща еще и гибкость, обеспеченная сетевыми возможностями стоящего за ним X-протокола. Конечно, попытки сделать что-то сумасшедшее типа проигрывания DVD по сети или запуска *Flash*-анимации – жесточайшие надругательства над сетью.

Протокол RFB, безусловно, дает пользователям выбор смешивать и согласовывать возможности клиентов и серверов, чтобы подобрать самую подходящую пару для своих целей. Это ваше дело, как назначить поведение вашего рабочего стола или функции сжатия, шифрования и упрощения для взаимодействия с нужными удаленными системами или рабочими станциями. Конечно, по большому счету, ваше решение будет определяться скоростью имеющегося интернет-соединения и временем, которое вы захотите потратить на настройку приложения. Как ни печально, при доступе к рабочему столу KDE на PDA с использованием GPRS, даже с самым сильным сжатием, продуктивно не поработаешь. LXF

К читателю

Скоро ли кто-нибудь реализует функцию RFB в X11? Получат ли приложения встроенное шифрование и сжатие для своей отрисовки? А вдруг разработчики Linux, рехнувшись, примутся использовать RDP для удаленного доступа? Сообщите нам свое мнение об этом Сравнении на lxf.letters@futurenet.co.uk

Как работает удаленный доступ

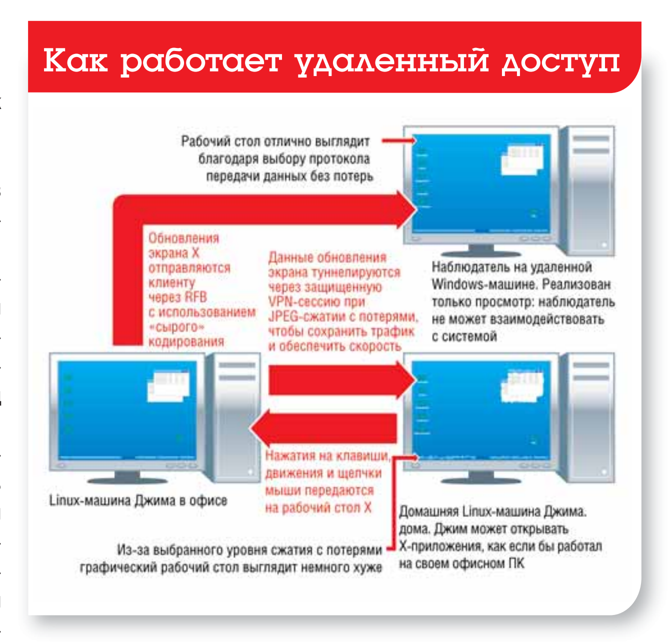


Таблица функций

Приложение	Тип	Протокол	Сжатие	Шифрование	Аутентификация	X-сервер	Окружение	Кросс-платформенность	Лицензия
Krdc	Интерфейс	RFB/RDP	☑	☒	Пароль	n/a	KDE	☑	GPL
NX	Клиент-сервер	NX	☑	☑	Пользователь и пароль	X11	X	☑	Проприетарная
TightVNC	Клиент-сервер	RFB	☑	☒	Пароль	Xvnc	X	☑	GPL
Tsclient	Интерфейс	RFB/RDP	☒	☒	Пароль	n/a	Gnome	☑	GPL
VNC	Клиент-сервер	RFB	☒	☒	Пароль	Xvnc	X	☑	GPL
X11	Сервер	X11	☑ [1]	☑ [1]	Ключ	X11	n/a	☑ [2]	Различные
X11vnc	Сервер	RFB	☑	☑	Пользователь и пароль	X11	n/a	☑	GPL

[1] Через SSH [2] Посредством инструментов сторонних производителей

Distrowatch

Обзор новостей разработки дистрибутивов Linux этого месяца.



ЛАДИСЛАВ БОДНАР
основатель, начальник
и сотрудник сайта
DistroWatch.com

Firefox против Opera

Недавний обзор Live CD Frenzy BSD на DistroWatch.com указал на «серьезный минус» этого – во всем остальном идеального – продукта: выбор *Opera* в качестве браузера по умолчанию. Как и следовало ожидать, это критическое замечание разожгло яростные баталии среди читателей. Некоторые согласны с автором в том, что открытый дистрибутив всегда должен держаться программ с открытым исходным кодом, а некоторые подчеркивают наличие у вышеупомянутого проприетарного браузера большого количества отличных функций.

Что касается использования, то здесь о сравнении речь просто не идет: пользователи *Firefox* составляют почти две трети от общего количества посетителей DistroWatch.com (который, как мы предполагаем, привлекает технически грамотную читательскую аудиторию). И только 8.5% читателей используют *Opera* для просмотра сайта.

Норвежская придумка

Тем не менее, *Opera* – слишком интересная программа, чтобы ее можно было так просто игнорировать. Во-первых, она меньше и быстрее, чем *Firefox*. Во-вторых, на моей системе Ubuntu *Opera* показывает куда большую стабильность, чем любой браузер на движке *Gecko*. И в-третьих, создатели этого норвежского браузера всегда демонстрировали весьма высокую целеустремленность; те, кто использует *Opera* несколько лет, должны помнить, что это был первый браузер с вкладками, первый браузер, который поддерживал «мышиные жесты», первый браузер с поддержкой text-to-speech, а сейчас – это первый браузер с виджетами и поддержкой BitTorrent.

И хотя я по-прежнему предпочитаю открытый *Firefox*, а не проприетарную *Opera*, вполне очевидно, какой браузер внес больше инноваций в мир интернет-серфинга. Хотите знать, какие потрясающие новые функции появятся у вашего браузера через годик-другой? Посмотрите на сегодняшнюю Opera – и поймете.

ladislav.bodnar@futurenet.co.uk

Будем проще

Xandros Desktop 4 Инструменты перехода с Windows на Linux и функции безопасности: решение для обычных пользователей?

Крупные релизы этого года от SUSE, Fedora Core и Ubuntu подарили энтузиастам свободного ПО новые потрясающие операционные системы. Однако появляется и другая группа дистрибутивов, которые куда меньше внимания обращают на «ух ты!» и куда больше – на «как».

Будучи либо проприетарными, либо содержащими проприетарные модули, эти дистрибутивы созданы с учетом интересов обычных пользователей: они объясняют, как установить, как пользоваться и как обновлять ваше ПО для рабочего стола. В эту группу входят Linspire и Turbolinux, но Xandros, в своем последнем релизе, зашел куда дальше большинства в создании интуитивно понятной операционной системы для новичков. Восстав из пепла Corel Linux, Xandros вполне сможет взлететь выше благодаря отлично отшлифованному домашнему дистрибутиву Xandros Desktop 4.



› *Xandros Networks*: безболезненный способ добраться до сети Linux.

Аккуратный дизайн

Что касается дизайна пользовательского интерфейса, это – один из самых дружественных дистрибутивов на сегодняшний день: его терминология покажется привычной любому новообращенному линуксоиду. Например, многие популярные приложения Linux переименованы, чтобы ясна была их цель: *DigiKam* превратился в Photo Manager, а *Amarok* стал Music Manager. Xandros Security Suite включает антивирусную программу, советующую почаще проводить сканирование на предмет наличия вирусов, а функция «product activation» («активация продукта»), необходимая для доступа к серверам обновления Xandros, укрепляет ощущение, что Xandros Desktop 4 создан не только для высоколобых ветеранов Unix.

> «Зашел дальше многих, создав ОС, интуитивно понятную новичку.»

Основанный в 2001 году, Xandros не пользовался особо славной репутацией в сообществе Linux. Фирма делала упор на пользователей Windows – недавняя реклама предлагала «оставшимся без поддержки пользователям Windows 98 и ME» 50% скидки при смене дистрибутива, а программы типа *CrossOver Office* и *Versora* включены в пакет, чтобы помочь беженцам с Windows обосноваться в новой среде.

Тем не менее, последний продукт Xandros обладает такими функциями, которые оценят и продвинутые пользователи. Xandros Security Suite – хороший пример утилиты, предоставляющей весьма полезный набор программ и мастеров конфигурации – не только для антивирусной программы *ClamAV*, но и для обновлений антивирусных баз, брандмауэра и программы защиты системных файлов; таким способом Xandros проверяет наличие в системе рут-китов и прочих «паразитов».

Наличие в установке по умолчанию программ и драйверов, не являющихся свободно распространяемыми, например, графических драйверов Nvidia и ATI, модулей *Flash Player* и *RealPlayer*, может отпугнуть от нее поборников чистоты, но дистрибутив заслуживает внимания. В версии 4 Xandros весьма продвинулся в своем стремлении создать хорошо интегрированный и дружественный к пользователю дистрибутив с отличным набором программ. Так что если вам нужна операционная система, основанная на Linux, для друга или члена семьи без особых технических наклонностей, на этот дистрибутив стоит взглянуть.

www.xandros.com

› Энди Ченнел «сделает» Xandros Desktop 4 в *LXF85*.

Решения Novell

SUSE Linux Enterprise Desktop 10 Этот коммерческий дистрибутив предназначен для бизнеса.

SUSE Linux 10.2 приняли со смешанными чувствами, но его коммерческий вариант Novell SUSE Linux Enterprise Desktop (SLED) 10 – это совсем другая история.

Без сомнения, SLED 10 станет одним из самых значимых релизов в истории дистрибутивов Linux. Причин на то несколько. Во-первых, Novell стал влиятельным и преданным сторонником Linux, к прямой выгоде всего сообщества. Во-вторых, за последние годы фирме пришлось побороться за существование, и если удача не вернется к ней – причем быстро – то есть риск, что её акционеры начнут придираться к стратегии Linux, которой придерживается Novell. И, в-третьих, Novell переживает крупномасштабный переход с Windows на Linux, и этот опыт, без сомнения, содействовал формированию SLED 10. Другие фирмы, которые предпримут попытку аналогичного перехода в будущем, смогут воспользоваться опытом Novell с пользой для себя.

В самом деле, SLED – не просто более лощеная, коммерциализированная версия SUSE Linux. Это – сочетание лучших программ с открытым исходным кодом, инструментов взаимодействия с Windows и революционных идей, обусловленных многочасовым изучением эргономичности. Опробовав его, даже самый убежденный скептик Linux будет вынужден признать, что Novell создал серьезного претендента на роль лучшей ОС рабочего стола, доступной на любой платформе – потенциального киллера для Vista.

www.novell.com/linux

› Мы вплотную займемся SLED 10 в **LXF85**.



› Улучшенное меню Computer и центр управления – лишь две из множества новых функций SLED 10.

BLAGoe дело

Blag Linux и GNU 50000 Клон Fedora, непохожий на оригинал – в нем только один CD!

Массе Linux-разработчиков Fedora Core дала отличную базу для создания специализированных программных решений. Добрая молва идет о таких из них, как BLAG Linux and GNU – проекте, разработанном Джеффом Мо [Jeff Moe] из Великобритании в попытке ответа на критику, которой подвергают Fedora за большой размер и отсутствие поддержки мультимедиа. BLAG противопоставляет этой критике добротный дистрибутив на одном CD, включающий Gnome, полную поддержку воспроизведения MP3 и DVD, немало отличных мультимедийных программ и других полезных инструментов из сторонних репозитариев для Fedora.

Среди прочего, BLAG включает офисный пакет Gnome (с *AbiWord* и *Gnumeric*), настольную издательскую программу *Scribus*, *Kiax*, виртуальный телефон и создатель наклеек *Glabels*. Видеоманов порадует коллекция видеоредакторов – в их числе *Kino* и *Cinerella*. Не меньшая радость ждет и любителей аудио, которые найдут здесь программы воспроизведения и декодирования аудио (*Audacity*, *Grip*), а также инструменты потокового аудио (*Muse*, *Streamtuner*). Завершает этот отличный дистрибутив масса приложений для Интернета и программы для работы с графикой.

Нечего и удивляться, что в качестве установщика системы BLAG использует Anaconda и предоставляет все обычные утилиты Red Hat/Fedora, превосходное определение устройств, а также обширное сообщество пользователей, всегда готовых помочь, с активными форумами и списками рассылок. LXF

www.blagblagblag.org



› BLAG, с его множеством программ редактирования аудио и видео, сущий рай для энтузиастов мультимедиа под Linux.

Частота появления новых релизов

Разница в частоте появления новых релизов у разных дистрибутивов огромна. Некоторые выпускают новый альфа- или пробный релиз каждые две недели, а некоторые – несколько новых версий в год. Приводимая таблица демонстрирует количество тестовых и стабильных релизов за 12 месяцев, с 1 июля 2005 по 30 июня 2006 г. Лидирует Mepis, благодаря большому количеству пробных релизов для разных редакций дистрибутивов. С другой стороны, Slackware выпустила только один релиз – версию 10.1 в сентябре 2005 г.



› Команда Уоррена Вудфорда умудряется выпускать новый релиз Mepis каждые две недели!

Дистрибутив	Число релизов	Комментарий
SimplyMepis	23	Специализируется на пробных релизах
SUSE	22	Релизу 10.1 предшествовало 16 альфа/бета/релиз-кандидатов
Damn Small Linux	22	Новая стабильная версия каждые два месяца
Ubuntu	20	«Летучие» и специализированные издания (сервер, SPARC)
FreeBSD	17	Параллельная разработка серий 5.x и 6.x
Mandriva	11	Снизила активность после перехода на годовой цикл выпуска релизов
Fedora Core	5	По словам Red Hat, трех пробников вполне достаточно
Gentoo	3	Beta? Что это?
Knoppix	3	Два релиза в неделю – это уже в прошлом
Debian GNU/Linux	2	«Многолетняя стабильность» (с периодическими обновлениями)
Slackware	1	Отсутствие новостей – хорошая новость [популярная английская пословица, – прим. пер.]

Хит-парад дистрибутивов

Десять самых посещаемых страниц дистрибутивов за июль (среднее количество визитов в день)

	Дистрибутив	Количество визитов	
1	Ubuntu	2,298	‹›
2	SUSE	1,495	‹›
3	Fedora Core	1,143	‹›
4	SimplyMepis	1,012	‹›
5	Damn Small Linux	863	‹›
6	Mandriva	795	‹›
7	Debian GNU/Linux	638	↑
8	PCLinuxOS	609	↑
9	Slackware	532	↑
10	Knoppix	526	↓

» DistroWatch.com определяет популярность дистрибутивов, основываясь на количестве посещений страниц, посвященных каждому из дистрибутивов. Хотя эти цифры и не отражают реального количества установок, они все же являются индикатором популярности того или иного дистрибутива в определенный период.

Прорыв в виртуальность



Аппаратная виртуализация позволяет запускать различные операционные системы на Linux одновременно, с высокой скоростью и бесплатно. Говорят об этом все; **Пол Хадсон** покажет вам, как сделать это.

Linux — это выбор: вы можете выбрать рабочий стол, выбрать браузер и даже выбрать логотип, который будет появляться при загрузке. Но основной выбор, без сомнения, самый сложный: какой дистрибутив лучше всех? Если вы похожи на нас, то, вероятно, скачете между дистрибутивами так же быстро, как Майк успевает выкладывать их на диск *Linux Format*; но настает время в вашей жизни, когда вам надоест делать переформатирование и вы начнете искать более разумный способ экспериментировать с новыми ОС. Этот способ – виртуализация: запустите множество ОС на одном компьютере и переключайтесь между ними с той же легкостью, как между программами.

Идея виртуализации не нова: *VMware* существует с 1999 года, и мы опубликовали две большие статьи про *Xen* за последние два года (*LXF67* и *LXF77*). Теперь, впрочем, производители процессоров создали виртуализацию на кристалле (on-chip). Работа ОС в виртуальной машине (ВМ) будет явно медленнее, чем на чистом компьютере, так что если вы сможете заставить процессор выполнять эту работу, весь процесс ускорится – по крайней мере, теоретически. Конкурирующие решения от Intel (VT или Virtualisation Tethnology, технология виртуализации) и AMD (SVM, или Secure Virtual Machines, безопасные виртуальные машины) начали поставляться с их процессорами, и наконец-то предприятия, любители новшеств и экспериментаторы среди вас могут попробовать виртуализацию, поскольку для этого она и предназначалась.

> «Конкурирующие решения от Intel и AMD стали поставляться в процессорах.»

Прорыв

Аппаратная виртуализация может, наконец, решить простую, но неприятную проблему: процессоры x86 не умеют выполнять виртуализацию хорошо. От выпуска 386-го до последнего, основанного на Core 2, чипа Intel изменилось немногое – набор базовых инструкций тот же, то есть код, скомпилированный для 386, также будет работать на новейшем оборудовании от Intel и AMD. На самом деле новые функции, например, MMX, SSE, двойные ядра и 64-битная обработка, просто были «прикручены» к существующей архитектуре x86, чтобы не обесценивать вложения пользователей и предприятий.

Но это также означает, что сохранились старые недостатки. До VT и SVM все чипы x86 использовали четыре уровня (кольца) привилегий для исполнения. На уровне 0 работали наиболее привилегированные процессы (обычно ОС), а на уровне 3 – наименее привилегированные (обычно приложения пользователя). Это способствовало безопасному разделению привилегий на аппаратном уровне: только приложения уровня 0 (нулевого кольца) могут напрямую управлять оборудованием, что теоретически не позволяет приложению выполнить что-то фатальное для машины. Но на практике приложениям часто нужен доступ к оборудованию, так что они нуждаются в вызовах ОС и запрашивают ресурсы, например, побольше памяти или отрисовку трехмерной графики.

Традиционная виртуализация работает путем запуска гостевой ОС поверх вашей основной ОС, в более высоком, менее привилегированном кольце. Когда гостевая ОС пытается выполнить привилегированную инструкцию, ее должен перехватить монитор виртуальной машины (VMM) и подменить ее, чтобы гостевая ОС не обнаружила, что работает не на чистом оборудовании. Эта техника – известная как двоичная трансляция – сложна и сопряжена с проблемами, в частности, самомодифицирующегося и самопроверяющегося кода. Кроме того, это довольно медленно, поскольку предполагается, что VMM постоянно наблюдает за инструкциями, проносящимися мимо, пока не сможет выстроить кэш.

Xen решил эту проблему, применив паравиртуализацию, когда исходный код гостевой ОС модифицируется так, чтобы она осознавала себя в виртуальной машине. Если гостевой ОС требуются дополнительные привилегии, VMM не будет выполнять никакой двоичной трансляции, поскольку гостевая ОС для обслуживания своих потребностей просто вызовет *Xen*. Это делает работу виртуализации намного быстрее, а хост- и гостевая системы связаны гораздо более тесно.

Аппаратная виртуализация устраняет эту проблему раз и навсегда, за счет создания нового набора уровней привилегий, которые размещаются позади традиционных уровней. Новые уровни (известные как VMX Root) становятся реальными привилегированными уровнями, в то время как старые уровни (известные как VMX Nonroot) виртуализируются чипом и используются там, где работают гостевые ОС. Когда гостевая ОС спрашивает «на каком уровне я нахожусь?», чип может честно ответить «уровень 0», потому что он действительно работает на уровне 0 – только это нулевое кольцо виртуальное. Таким образом, все гостевые системы могут работать, не прибегая к двоичной трансляции, поскольку они не замечают разницы между уровнем 0 реальной машины и уровнем 0 внутри VMX Nonroot.

Поскольку это передовая технология, аппаратная виртуализация поддерживается не всеми чипами. В случае с Intel все новые чипы, основанные на Core 2, поддерживают аппаратную виртуализацию VT. Некоторые чипы постарше также работают, но подавляющее большинство чипов Pentium 4 не совместимы с VT. Что касается AMD, то

Зачем виртуализировать?

Для домашних пользователей: Сыты по горло переустановкой дистрибутивов? Запустите один маленький дистрибутив как Domain-0, затем виртуализуйте поверх него любые другие дистрибутивы.

Для программистов: Нужно протестировать вашу программу на различных дистрибутивах, операционных системах и настройках? Легко: создайте столько виртуальных машин, сколько у вас платформ, затем разработайте набор тестов для запуска по ночам.

Для администраторов серверов: Застряли на 25%-ной загрузке сервера? Используйте виртуальную инфраструктуру для снижения времени простоя, увеличения гибкости и моментального предоставления ресурсов для всех ваших задач.

Alex Williamson

только новейшие чипы, основанные на AM2, поддерживают технологию Secure Virtual Machines. Линейка Opteron должна обновиться в конце этого года, но сейчас SVM поддерживают только новые Athlon 64 и Turion 64.

Интересно заметить, что хотя VT и SVM решают одну и ту же проблему, они работают совершенно по-разному: чипы AMD64 имеют интегрированный контроллер памяти, а Intel держит его на северном мосту. На практике это означает, что чипы Intel должны виртуализовать контроллер памяти программно, тогда как AMD может делать это аппаратно, обеспечивая еще больший потенциал для повышения скорости виртуализации.

Да, Xen все еще нужен

Размещение поддержки виртуализации на процессоре не отменяет потребности в программном обеспечении типа *VMware* или *Xen*, потому что все, что находится за пределами процессора (и контроллера памяти, в случае с AMD), по-прежнему нужно виртуализовать. Видеокарта, звуковая карта, жесткие диски и т.д. нуждаются в программной обработке, и здесь системы виртуализации все еще при деле. Наряду с *VMware* и *Xen* вы можете также использовать Qemu и *Parallels Workstation*, но хотя вся четверка предоставляет полную эмуляцию оборудования, кроме процессора, Qemu не имеет никакой поддержки технологий виртуализации Intel и AMD, и его производительность страдает. »

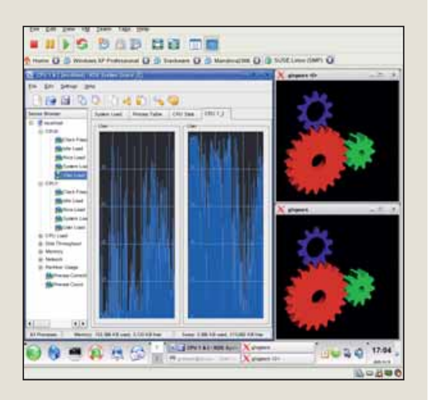
Xen против VMware

Наиболее явное отличие между *Xen* и *VMware* в том, что *Xen* устанавливает себя как основную ОС компьютера и загружается, когда включается компьютер. Когда ОС *Xen* запустится, она создаст свою первую виртуальную машину с операционной системой, известную как Domain-0. Она имеет особые привилегии доступа к оборудованию и имеет полное управление над любыми гостевыми системами, вызываемыми после нее.

В *VMware* ваша первоначальная ОС остается неизменной – *VMware* запускается как приложение поверх исходной ОС, переключая ее в режим хоста. Новые виртуальные машины создаются поверх хост-ОС как гостевые, и запускаются внутри приложения *VMware*.

Какая из этих двух идей лучше – вопрос спорный; но мы склоняемся в сторону подхода *Xen*, поскольку если в хост-ОС на *VMware* обнаружатся проблемы (или она рухнет), все виртуальные машины «завалятся» вместе с ней.

Между прочим, профессиональная версия *VMware* – *VMware ESX Server* – работает во многом подобно *Xen*, так что, возможно, люди *VMware* тоже считают это лучшим подходом!



› ***VMware*** **позволяет эмулировать многопроцессорный компьютер на однопроцессорном.**



Попробуйте сами

Это будет серьезной технологией 2007 года – испытайте ее сегодня и приплюсуйте к своему счетчику крутизны солидную добавку.

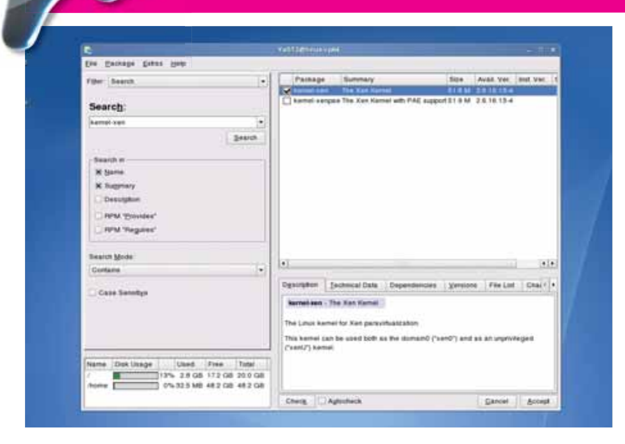
Есть чип, поддерживающий VT/SVM? Отлично! Вы в нескольких минутах от запуска Windows в Linux при помощи аппаратной виртуализации. Это можно сделать двумя способами: через *Vmware* — тогда виртуализация получится мгновенно, но это закрытая программа, не позволяющая модифицировать аппаратную часть во время работы; или через *Xen*, который полностью открыт, поставляется включенным в ряд основных дистрибутивов, но потребует некоторых усилий, чтобы заставить его работать. Хотя мы пытались заставить *Parallels Workstation* работать с VT, это почти всегда заканчивалось зависанием — приглашаем вас попробовать самим, и если вы заставите его работать, сообщите нам.

Нашими тестовыми машинами с *Xen* были двухпроцессорный Intel Xeon 5160 (итого четыре ядра), с 8 ГБ памяти, и AMD Athlon 64 X2 5000+ с 1 ГБ памяти. На момент написания статьи новый чип Xeon 5160 на базе Woodcrest/Core 2 был старшим серверным чипом от Intel, а машина AMD ориентировалась на домашних пользователей – все результаты тестирования, которые здесь напечатаны, лучше использовать для сравнения одной машины, работающей под «чистой» ОС, с той же машиной под управлением виртуализованной ОС, а не сравнивать Intel и AMD.

Наша тестовая платформа

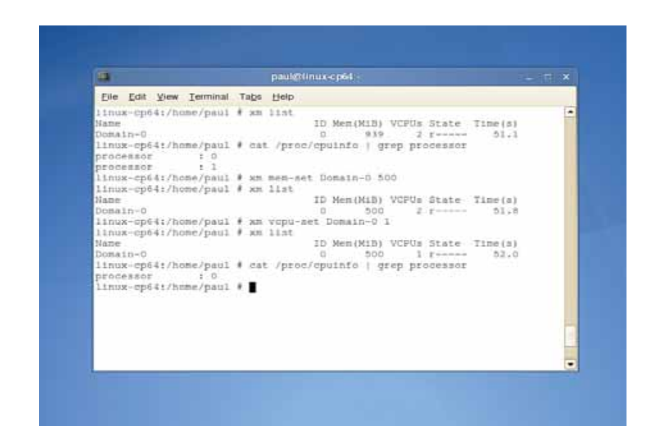
Мы установили SUSE 10.1 на обе машины, используя 32-разрядные хосты и 32-разрядные гостевые ОС. SUSE 10.1 – превосходный выбор для испытания *Xen*, поскольку она поставляется с пакетами ядра Xen, устанавливаемыми через *Yast*. Также здесь есть специальный графический интерфейс для установки и настройки виртуальных машин, но на момент написания статьи он не работал – так что мы делали все вручную.

Шаг за шагом:



1 Установите пакеты Xen

Из *Yast*, поищите «kernel-xen», и вы увидите два пакета: **kernel-xen** и **kernel-xenp-ae** (*см. ниже врезку «Обратите внимание»*). Вам также понадобятся **cloop-kmp-xen**, **xen**, **xen-libs**, **xen-tools** и **xen-tools-ioemu**, и любые дополнительные драйверы, какие пожелаете.



4 Настройте работу с ОЗУ

Запустите **xm list** для получения информации о Domain-0. Вы можете использовать **xm mem-set Domain-0 500**, чтобы назначить ему 500 МБ памяти. Можете также использовать **xm vcpu-set Domain-0 1**, чтобы задать ему один процессор. Изменения вступят в силу немедленно, так что вы сразу увидите, если памяти не хватает!

Выявление неисправностей

Xen на VT/SVM сейчас довольно сложен – мы обнаружили многочисленные зависания, падения и ошибки, пока писали эту статью. Обычно это ошибка пользователя, но *Xen* любезно умалчивает об этом. Вот решение некоторых из общих проблем:

Проблема	Решение
Система зависает	Достаточно ли свободной памяти, чтобы создать виртуальную
«Имя ВМ уже использовано»	Каждая ВМ должна иметь уникальное имя в своем конфигурационном файле.
XP не загружается	Убедитесь, что «boot» в конфигурационном файле установлен в «d».
Графика повреждена	Установите «stdvga» в 1 в конфигурационном файле. Когда XP установится, вы можете снова поменять его на 0.
Гостевая ОС не может прочитать более 4 ГБ памяти	Установите «pae» в 1 в конфигурационном файле.
«Платформа не поддерживает HVM»	Ваш чип с функцией VT/SVM? Эта функция запрещена в вашем BIOS? Заметьте: когда изменяется эта опция BIOS, вам нужно полностью отключить питание системы.

Обратите внимание

Windows XP на *Xen* работает, только если ваш процессор входит в следующий список:

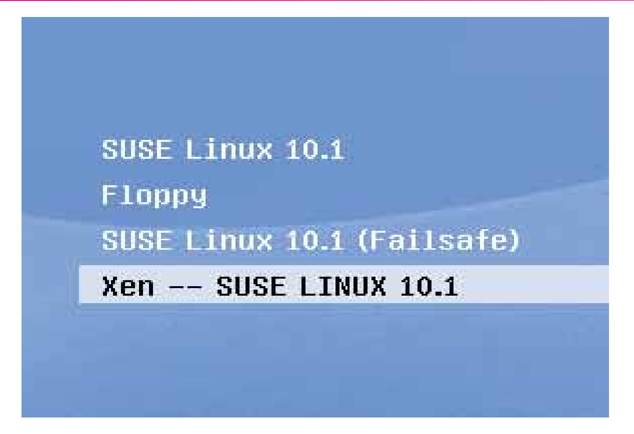
» Intel:
Core Duo
Core 2 Duo
Pentium Extreme Edition 955/965
Pentium D 920/930/940/950/960
Xeon 5030/5050/5060/5063/5080/5160/7020/7030/7040/7041

» AMD:
AM2-based Athlon 64s
AM2-based Athlon 64 X2s
Athlon 64 FX-62

Имеете ОЗУ больше 4 ГБ?

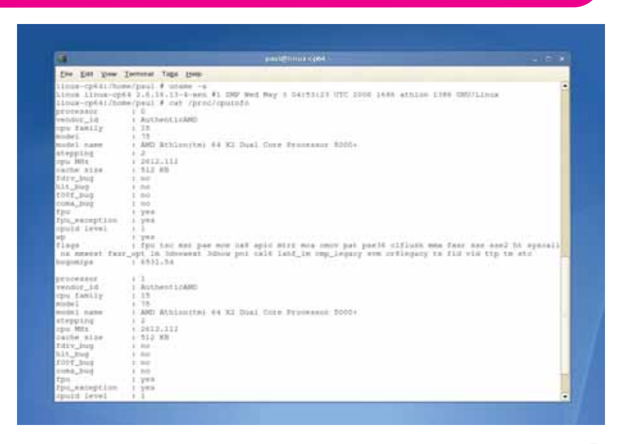
Если ваша машина имеет более 4 ГБ памяти, вы, вероятно, уже используете Linux-ядро с PAE (Physical Address Extension). Некоторые дистрибутивы именуют такие ядра по типу linux-kernel-pae или kernel-bigsmp. Устанавливая свое ядро *Xen*, используйте также и ядро PAE *Xen*, иначе ваши гостевые системы в Xen не смогут получить доступ более чем к 4 ГБ ОЗУ.

запускаем VT- или SVM-виртуализацию с помощью Xen



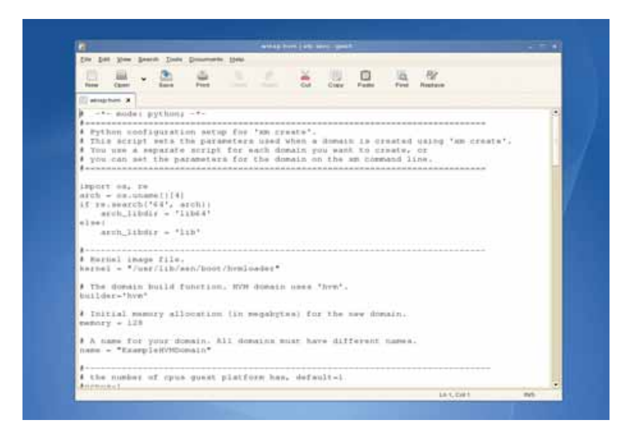
2 Перезагрузитесь

Перезагрузите свой компьютер и выберите Xen – SUSE LINUX 10.1, когда появится загрузочное меню *Grub*. Пролетит кое-какой текст, и ваш компьютер загрузится как обычно. Не беспокойтесь, если исчезнет графическая заставка вашей SUSE: это не является дурным знаком.



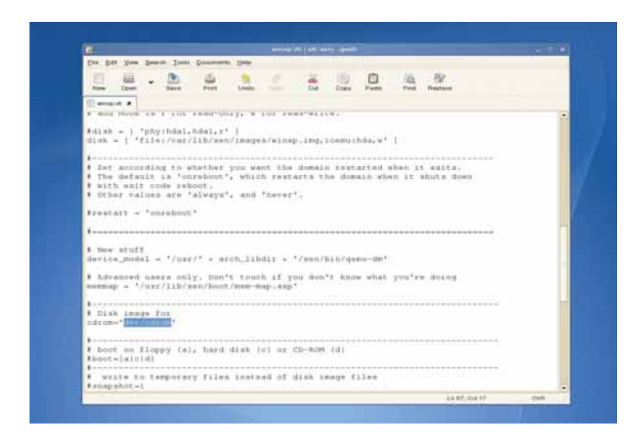
3 Проверьте установку

Поздравляем – вы загрузились в *Xen*! Вы можете проверить это, открыв консоль и запустив команду **uname -a**. Ваша версия ядра должна теперь походить на 2.6.16.13-4-xen.



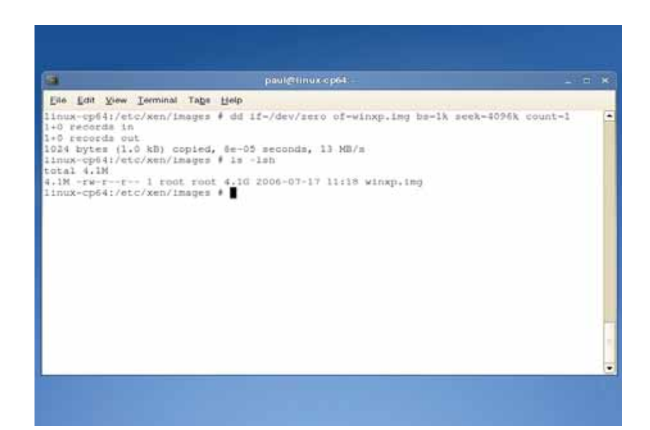
5 Откройте файл конфигурации

Из командной строки скопируйте **/etc/xen/examples/xmexample.hvm** в **/etc/xen/winxp.hvm**, затем откройте **winxp.hvm** в своем любимом редакторе от имени root. Мы должны сконфигурировать его, чтобы запустить Windows.



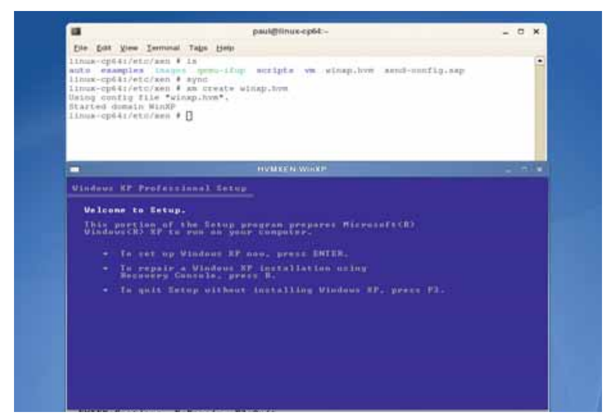
6 Настройте XP

Установите «memory» в значение в МБ для этой ВМ, а «name» – в уникальное значение, хотя бы «WinXP». Отредактируйте строку «disk» так, чтобы он указывал на **winxp.img**. Установите «cdrom» в значение **/dev/cdrom**, «boot» в **d**, «sdl» в **1**, «vnc» в **0** и «vncviewer» в **0**. Удалите любые символы # из начала этих строк.



7 Создайте виртуальный жесткий диск

Отведите вашей системе XP пространство на диске, переключившись в каталог **/etc/xen/images** и выполнив команду **dd if=/dev/zero of=winxp.img bs=1k seek=4096k count=1**. Она создаст виртуальный диск размером 4 ГБ, который можно будет смонтировать как loopback-устройство внутри *Xen*.



8 Финал: Windows на Linux!

Теперь выполните команду **xm create winxp.hvm** из каталога **/etc/xen**, и – если все будет нормально – появится окно с запущенным инсталлятором Windows. Будьте осторожны: этот шаг самый рискованный из всех; рекомендуем сперва выполнить команду **sync** и убедиться, что буфер вашей файловой системы очищен [также подразумевается, что установочный диск Windows XP вставлен в CD-ROM, – прим. ред.].

»



аппаратная виртуализация?

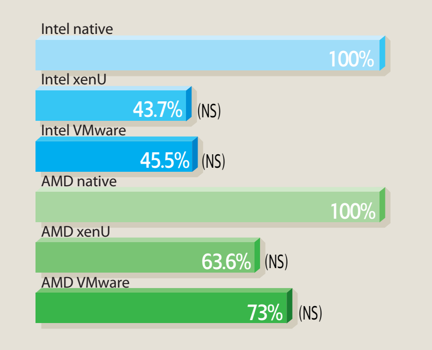
Легко утверждать, что «это» быстрее, чем «то» – но вот холодные, неоспоримые факты.

Получить графики испытаний аппаратной виртуализации – задача не простая: фактически, потребовалось две недели на установку, настройку, переустановку, копирование и тестирование на обеих наших машинах, чтобы получить то, что мы считаем окончательными результатами. Кроме того, не забывайте, что реально мы можем сравнивать только Intel с Intel, а AMD — с AMD, поскольку эквивалента AMD64 для поддерживающего VT чипа Xeon пока нету. Более того, большинство этих тестов автоматически получают преимущества от дополнительных ядер там, где они доступны, так что баллы Intel будут с учетом четырех ядер по сравнению с двумя ядрами Athlon 64.

Мы провели в общей сложности восемь тестов, используя шесть сценариев и две машины. Каждый тест на каждой машине запускался на «родной» ОС, в гостевом домене *Xen* (xenU) и внутри виртуальной машины *VMware*. Поскольку нас особенно интересовала аппаратная виртуализация для этих функций, большинство тестов использовало Windows XP в качестве гостевой ОС, за исключением теста кодирования Ogg: тут использовалась SUSE 10.1.

Помимо установки монитора виртуальной машины и гостевой ОС, выполнялась небольшая дальнейшая настройка системы. Как для *VMware*, так и для *Xen* выделялся одинаковые объем ОЗУ и место на диске для работы, и от обоих требовалось предоставить виртуальную сетевую карту; обоим дана свобода управлять процессорами как они сочтут нужным. Одно небольшое исключение было сделано для *Xen*, запускаемого на машине Intel, поскольку Intel VT, похоже, имеет проблемы с чипсетом Cirrus Logic, эмулируемым *Xen* (*см. врезку «Заставь работать VT-графику»*).

Интерпретация результатов



Сравнивать «чистый» Intel с «чистым» AMD несправедливо: чип Intel имеет в два раза больше ядер и в восемь раз больше ОЗУ. Но мы можем рассмотреть различие в скорости между чипами Intel и AMD, когда они работают виртуально. Средняя скорость чипа Intel, управляемого гостевой ОС через Xen, составила 43,7% от «чистого», по сравнению с 45,5% на *VMware*. На AMD результаты поднялись до 63,6% для Xen и 73% для *VMware*.

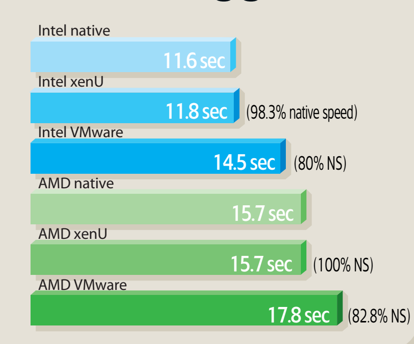
Это отчасти объясняется интегрированным в AMD контроллером памяти: тогда нужно программно реализовывать одним устройством меньше. Но другой фактор, заслуживающий внимания – тот, что тесты используют конфигурацию «из коробки», и оба приложения управляют ресурсами гостевых ОС в свободном полете. В случае с Xen вы можете, например, отдать виртуальной гостевой ОС весь процессор, и многопоточным тестам будет куда вольготнее.

Кодирование Ogg

Выполнялось: Конвертировался файл MP3 (6 МБ) в формат Ogg с качеством 9 (наивысшим).

Проверялось: Нагрузка на CPU;

чем ниже, тем лучше



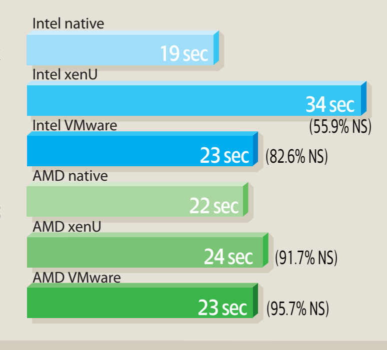
Вердикт Это одна из областей, где *Xen* должен быть на коне: была бы производительность процессора высока, а *Xen* просто работает как тонкая прослойка между гостевой ОС и хостом.

Запуск Windows

Выполнялось: «Холодный» запуск Windows XP.

Проверялось: Смесь CPU, ОЗУ и жесткого диска – прекрасный тест реальной ситуации;

чем ниже, тем лучше

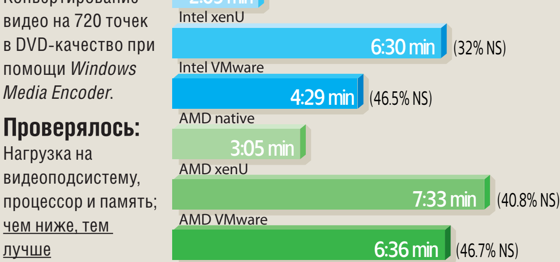


Вердикт Неудача постигла VT от Intel, разница в скорости по сравнению с «чистым» оборудованием – почти вдвое. На AMD разница минимальна. Только *VMware* хорошо держит темп на обеих платформах.

Кодирование Windows Media

Выполнялось: Конвертирование видео на 720 точек в DVD-качество при помощи *Windows Media Encoder.*

Проверялось: Нагрузка на видеоподсистему, процессор и память; чем ниже, тем лучше

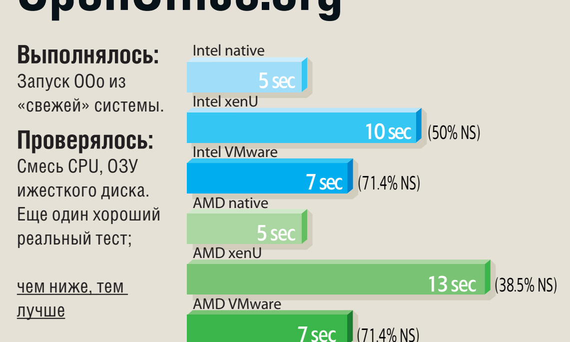


Вердикт *Xen* на Intel дает менее 33% от скорости чистого оборудования Intel, в основном из-за слабой эмуляции графики. AMD справился с тестом лучше, благодаря виртуализованному контроллеру памяти.

OpenOffice.org

Выполнялось: Запуск OOo из «свежей» системы.

Проверялось: Смесь CPU, ОЗУ и жесткого диска. Еще один хороший реальный тест; чем ниже, тем лучше

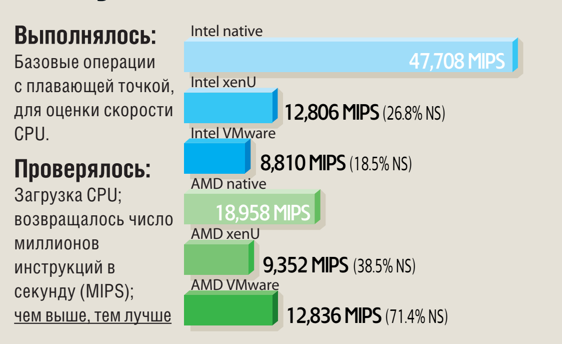


Вердикт Запускаясь в первый раз, *OpenOffice.org 2.0* буквально истязает жесткий диск, но *VMware* и здесь умудряется вывернуться с очень приличным результатом.

Dhrystone

Выполнялось: Базовые операции с плавающей точкой, для оценки скорости CPU.

Проверялось: Загрузка CPU; возвращалось число миллионов инструкций в секунду (MIPS); чем выше, тем лучше

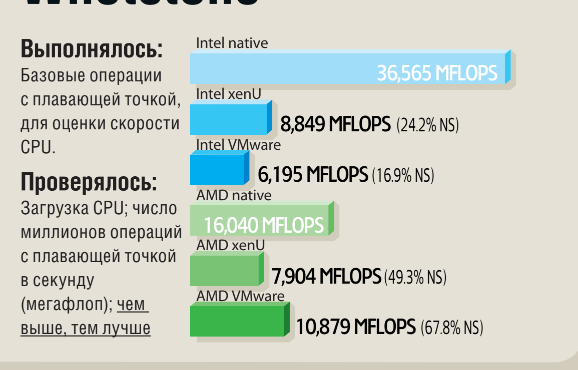


Вердикт Чистый Intel властвует в этой категории, но при виртуализации заметно резкое падение. Для сравнения, AMD при виртуализации замедляется не более чем на 50%.

Whetstone

Выполнялось: Базовые операции с плавающей точкой, для оценки скорости CPU.

Проверялось: Загрузка CPU; число миллионов операций с плавающей точкой в секунду (мегафлоп); чем выше, тем лучше

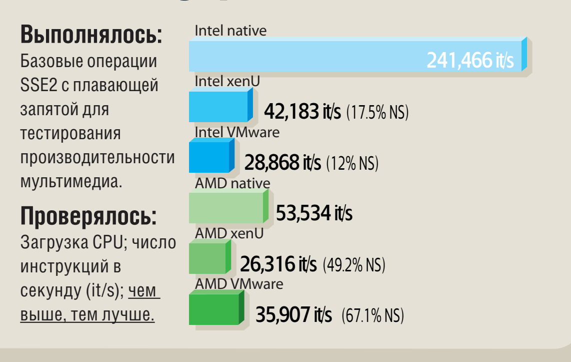


Вердикт Чип Intel превосходен при работе с «чистой» ОС, но отстает на виртуальных машинах. Результаты AMD близки к его результатам Dhrystone, лишь с небольшим отставанием.

Floating-point x4 iSSe2

Выполнялось: Базовые операции SSE2 с плавающей запятой для тестирования производительности мультимедиа.

Проверялось: Загрузка CPU; число инструкций в секунду (it/s); чем выше, тем лучше.

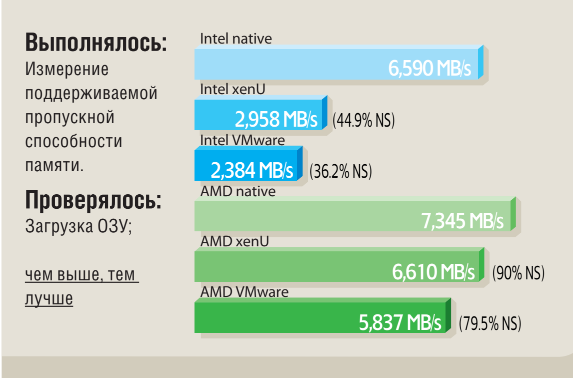


Вердикт Новое серверное ядро Woodcrest от Intel вызывающе быстро, но плохо уживается с виртуализацией. У AMD падение куда менее драматичное. Самые лучшие показатели здесь у *VMware*.

Пропускная способность ОЗУ

Выполнялось: Измерение поддерживаемой пропускной способности памяти.

Проверялось: Загрузка ОЗУ; чем выше, тем лучше



Вердикт AMD не намного обошел Intel в «чистом» режиме, но «сокрушил» его как под *Xen*, так и под *VMware*. Опять-таки, аппаратная виртуализация контроллера памяти – существенное отличие.

Заставь работать VT-графику

Наших тесты обнаружили, что реализация VT от Intel в паре с *Xen* вызывала искажение графики и даже падения, когда выполнялась загрузка с CD. Если вы столкнетесь с такой же проблемой, решением будет редактирование файла конфигурации вашей ВМ: измените строку «stdvga» на значение «1», чтобы принудительно включить стандартный низкоцветный режим VGA. Как только ваша гостевая ОС установится и будет настроена, вы можете изменить stdvga снова на 0, для возврата к эмуляции графической системы Cirrus Logic.

»

Лучше не бывает?

Следите за этой областью: впереди еще долгий путь, и аппаратная виртуализация будет становиться лучше и лучше.

Увлекательные сами по себе, наши диаграммы показывают, что неоптимизированная аппаратная виртуализация в ряде тестов отстает от ОС, запущенных в «чистом» виде. Конечно, теперь можно запустить любую ОС поверх другой ОС, но для первой попытки, производительность здесь еще недостаточная. Можно достичь хорошей скорости, вручную подстроив виртуальные машины и отведя им процессорные ядра, но это довольно утомительно — особенно сейчас, когда многоядерные архитектуры скорее стандарт, чем исключение. Вдобавок ручная оптимизация лишает виртуальную инфраструктуру свойственной ей гибкости. Поскольку *Xen* и *VMware* расширили свою поддержку для VT и SVM, в частности, для лучшей обработки многоядерных сценариев, мы надеемся, что наши цифры немного поднимутся, но в конечном итоге для достижения полноскоростной виртуализации нужно реформировать оборудование.

Как мы видели, виртуализуемый контроллер памяти – огромное подспорье для AMD, поскольку программно требуется эмулировать на одну вещь меньше. Виртуализация высокоскоростных устройств – ключ к снижению общей латентности системы, и неудивительно, что Intel уже работает над следующей версией VT, где появится штука, называемая Directed IO и разработанная, чтобы поддержать аппаратную виртуализацию мириадов наших обычных устройств ввода-вывода – клавиатуры, мыши и устройств USB.

> «Самые значительные новые разработки — в ядре Linux.»

При наличии поддержки аппаратной виртуализации ввода-вывода останется решить только одну проблему: нужна виртуальная графическая система, способная обеспечить гостевым машинам приличное ускорение графики. Сейчас большинство гостевых ОС привязаны к эмуляции графического уровня Cirrus Logic, которая ощутимо тормозила наши тесты по части графики. При такой слабой 2D-графике безнадежно запускать на гостевой ОС графически богатые приложения, так что трехмерное моделирование, игры и даже окружение рабочего стола уровня 2.5D/3D, например, *Xgl* и Windows Vista, не поддерживаются.

Пока Intel работает над следующим поколением VT, инновации происходят и в программном обеспечении, которое обеспечивает прогресс виртуализации. *Xen* 3.0 принес с собой огромные улучшения, включая поддержку VT и SVM, что позволило нам написать эту статью. Он также ввел приличную поддержку SMP, большую производительность памяти и порт для Itanium. Но есть кое-какие интересные вещи, все еще находящиеся на стадии разработки, которые могут увидеть свет в следующем крупном релизе *Xen*.

Большие скачки ПО

Список интересных новых функций возглавляет виртуализованная графическая подсистема, использующая ПО, а не оборудование. Чтобы обеспечить базовое ускорение графики, используются разделяемые буферы кадров, и есть шансы, что мы увидим даже виртуализованный OpenGL, доступный современным трехмерным графическим интерфейсам. Другая область, где, похоже, разра-



› **Чип Core 2 Duo от Intel с поддержкой VT превосходен для виртуализации на чем угодно, от ноутбука до сервера.**

ботки ведутся интенсивно – качество обслуживания: разработчики *Xen* хотят, чтобы виртуальные машины сами балансировали себя на доступных процессорах, а также эффективнее использовали ОЗУ.

На будущее у разработчиков *Xen* есть ряд исследовательских идей, которые они хотели бы реализовать в *Xen*, не последние из которых – детерминированное воспроизведение и интерпозиция ввода-вывода. Идея детерминированного воспроизведения – сделать легковесный снимок виртуальной машины в процессе работы, а затем вести журнал всего ввода и событий: если система рухнет, разработчик сможет вернуться к снимку, шаг за шагом проиграть журнал и выяснить, где произошла ошибка. Интерпозиция ввода-вывода – еще одна интересная идея для отладки, позволяющая разработчикам эмулировать неисправности на уровне ввода-вывода виртуальной машины, чтобы, например, смоделировать простой сети или большие задержки в доставке пакетов.

Следующий шаг

Возможно, самые значительные новые разработки имеют место в ядре Linux, а не в самом *Xen*. *Xen* очень близко подошел к слиянию с основным ядром, которое сделало бы *Xen* доступным в каждом дистрибутиве как стандарт. Сейчас дистрибутивы, поставляющие *Xen* – например, Red Hat и SUSE – должны включать его в свои ядра. Но планируется создание независимого от поставщиков интерфейса к виртуализации ядра, тогда *Xen*, *VMware* и вообще любой разработчик виртуализации смогут включить свою систему в ядро Linux. Чтобы это сработало, потребуется некоторая степень согласования между *Xen* и *VMware*, и обе команды близки к этому. Их конечные цели совпадают, как во многом совпадают и их коды, но потребуется некоторая работа, чтобы завершить объединение оставшихся 25% или около того, пока не будет готово совместное решение.

Пока не планируется вводить совместимость виртуальных машин *VMware* и *Xen*, что явилось бы логичным следующим шагом. Сейчас формат ВМ, используемый *Xen*, полностью отличается от используемого в *VMware*, то есть виртуальную машину, созданную в одном приложении, нельзя запросто использовать в другом.

Xen уже включен как стандарт в SUSE 10.1 и Fedora Core 5, и, вероятно, сейчас будет включаться и в другие дистрибутивы: *Xen* 3.0 достаточно стабилен, чтобы считаться свободным от ошибок. Марк Шаттлворт сказал, что он хочет видеть ряд захватывающих функций в следующем релизе Ubuntu, и мы думаем, что можно держать пари – *Xen* будет одной из них. Вдобавок и SUSE, и Red Hat обдумывают идею сделать *Xen* доступным по умолчанию в своих дистрибутивах, это даст возможность за пять минут устанавливать новые гостевые домены, поскольку машина уже будет виртуализована. Поскольку основной домен *Xen* (известный как *Xen*0) имеет больше привилегий по управлению оборудованием, расплачиваться за это скоростью практически не придется – и это действительно очень заманчивая перспектива.

Как только аппаратная виртуализация сумеет превысить 90% производительности по сравнению с «чистым железом» без ручной подстройки файлов конфигурации, уклоняться от виртуализации станет практически незачем. Установка ВМ для базы данных и ВМ для web-сервера на одном и том же оборудовании и передача динамического назначения требуемых процессоров и памяти на откуп *Xen* или *VMware* будет способствовать значительной экономии – и в серверном зале, и дома. Вам больше не понадобится машина для Linux, машина для OS X и машина для Windows, и использование сервера на 25% не будет больше считаться средним значением. Вместо этого мы сможем разместить все наши операционные системы на одном центральном компьютере дома и получать доступ к нашему рабочему столу в любом месте, где есть сетевое подключение. Обещания VT и SVM станут реальностью – в смысле, виртуальной реальностью. LXF

Как добиться максимальной производительности

Как мы видим, и *Xen*, и *VMware* борются за достижение максимальной производительности, когда работают на нескольких ядрах, но если вы готовы пожертвовать некоторой гибкостью, вы можете достичь максимума, вручную настроив ваши ВМ. Самое простое, что можно сделать – найти настройку **vcpus** в файле конфигурации *Xen* и изменить ее так, чтобы она соответствовала общему числу ядер на вашей системе. Если у вас машина с поддержкой HyperThreading, вам, возможно, захочется ради эксперимента удвоить число **vcpus** и выяснить, поможет тут HT или нет.

Итак, на одноядерном Athlon 64 установите **vcpus** в 1, а на двухядерном, двухпроцессорном поддерживающем HyperThreading Xeon начните с четырех **vcups** и затем попробуйте перейти на восемь, чтобы посмотреть, будет ли заметно повышение производительности. Заметьте, что HyperThreading обновилась с SSE3 («Новые инструкции Prescott»). Раньше некоторые приложения шли с HT быстрее, а некоторые – медленнее, но после обновления набора команд скорость в худшем случае просто не изменится.

› В наших тестах использовался новый основанный на Woodcrest сервер Magnetar X2 от Armari, и мы смогли достичь максимальной производительности, только переопределив установки по умолчанию *Xen*.

А как же Qemu?

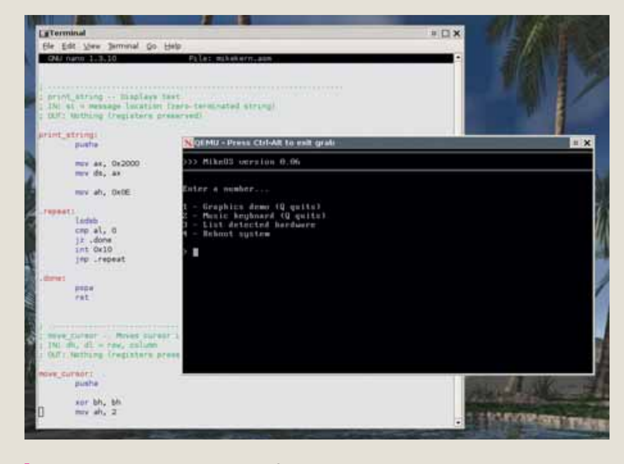
Читайте наше руководство на следующем развороте!

Qemu – превосходная и давно известная программа эмуляции, которая позволяет запускать Windows (или любую другую ОС) внутри виртуальной машины. Несколько лет она являлась приложением с открытым кодом, но поскольку она только эмулирует операционные системы (т.е. транслирует их программно), а не виртуализует их (т.е. запускает их на самом чипе), она всегда работала довольно медленно. Однако ее создатель реализовал модуль виртуализации, позволивший *Qemu* работать гораздо быстрее – с оговоркой, что это модуль закрытый.

Если вас не волнует моральное падение по отношению к свободному ПО, *Qemu* с бесплатным акселератором работает гораздо быстрее. Мы хотели было включить *Qemu* в наше тестирование, но уж очень странные вышли результаты (невозможные или маловероятные цифры). Тем не менее *Qemu* работает, причем неплохо. Например, *Xen* вовсю использует *Qemu* в режиме аппаратной виртуализации, потому что эмуляция устройств определенно реализована в нем лучше, чем где-либо.



› *Qemu* – отличный способ попробовать дистрибутивы с дисков Linux Format..



› Windows «гонять» можно было давно, но только сейчас для *Qemu* нашлось достойное применение: запуск MikeOS.

QEMU: руководство

Пока еще не купили один из новых процессоров AMD или Intel? Не печальтесь – *Xen* не единственное средство виртуализации, доступное в мире Open Source.
Пол Хадсон передает микрофон **Андрею Дубравину**.

Проверяя очередной раз свою электронную почту, я обнаружил письмо с сообщением об ошибке в разрабатываемом мной приложении. Я стал изучать полученную информацию и сделал вывод, что в моей рабочей ОС данной ошибки не наблюдалось. В чем же дело? Немного поразмыслив, я пришел к следующему выводу: в сообщении было указано, что мой респондент использовал Debian 3.1; я использовал эту же ОС, но с обновленными библиотеками. Можно было воспользоваться информацией об ошибке и исправить её, надеясь на то, что я все сделаю правильно и эта ошибка больше на появится. Кроме того, можно было послать новую версию программы пользователю, чтобы тот проверил её на своей системе, но согласится ли он? Могли обнаружиться и другие ошибки, связанные с несовпадением версий библиотек. Что же делать? Я принял решение поставить ещё одну ОС. Но мне не хотелось изменять таблицу разделов своего жесткого диска: это может повредить установленным ОС, да и кто знает, сколько их еще мне придется поставить? Я решил, что лучшим выходом будет воспользоваться виртуализацией, а именно – QEMU. Эта программа является свободной и распространяется с исходными текстами, исключение составляет модуль ядра *kqemu* (также известный как QEMU Accelerator), который используется для повышения скорости работы. Он доступен только в двоичном виде, его можно бесплатно использовать, но не распространять.

Ну что же, приступим! Для начала необходимо установить QEMU. Найти последнюю версию программы и другую информацию, касающуюся QEMU можно по адресу http://fabrice.bellard.free.fr/qemu/. Там же можно найти и модуль ядра *kqemu*. Этот модуль заметно повышает производительность, поэтому я советую его поставить, хотя QEMU будет работать и без него. В своих экспериментах я использовал ОС Debian 3.1 и QEMU версии 0.8.2.

Модуль *kqemu* может работать в двух режимах: for user code и for user and kernel code. Первый режим устанавливается по умолчанию, для установки второго режима необходимо использовать ключ -kernel-kqemu. Второй режим является более быстрым, но его использование может привести к некорректной работе некоторых ОС.

Что касается оперативной памяти, выделяемой гостевой ОС, то по умолчанию QEMU использует 128 Мб. Меня это число вполне устроило, но если вы хотите поменять это значение, то можете воспользоваться опцией **-m**, в качестве параметра этой опции передается размер ОЗУ в мегабайтах.

При дальнейшем изложении, в примерах команд, опции -kernel-kqemu и -m указываться не будут. Если в этом возникнет необходимость, вы можете добавить их самостоятельно.

Для начала попробуем запустить какой нибудь LiveCD прямо из текущей ОС. В качестве оного можно взять один из дисков LXF. Команда для загрузки виртуальной машины прямо с CD выглядит следующим образом:

```
qemu -cdrom /dev/cdrom
```

Что может быть проще? После того, как выполнится эта команда, перед вами откроется окно QEMU, играющее роль дисплея виртуальной машины. Загруженный в QEMU LiveCD Games Knoppix представлен на **рис. 1**.

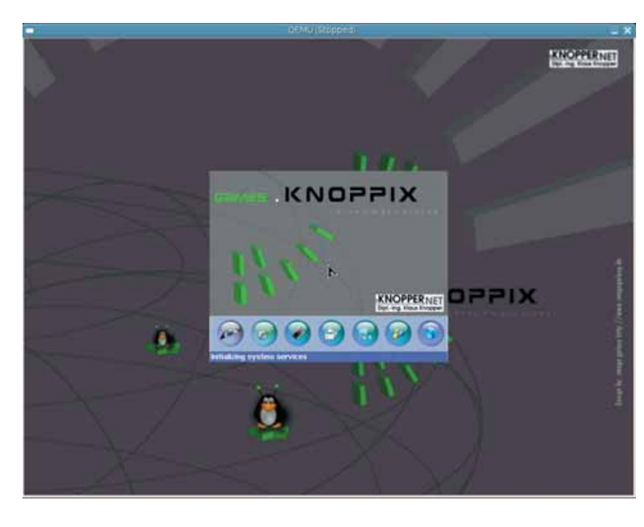
Теперь мы можем изучать LiveCD и при этом слушать свою любимую музыку. Что же дальше? Загружаться с установочного CD мы научились, но нам еще необходимо выделить место для самой ОС. В нашем случае ОС будет устанавливаться в образ диска. Существует несколько форматов для хранения образов, которые поддерживает программа QEMU, мы воспользуемся самым распространённым – **raw**.

Мне известно два способа создать пустой образ: с помощью утилиты *qemu-img* из состава QEMU и вручную, посредством утилиты *dd*. С использованием *qemu-img* можно создать образы любых форматов, поддерживаемых QEMU, утилита *dd* позволяет создать только образы формата **raw**. Если быть более точным, то с помощью *dd* можно создать заполненный нулями файл, который и является пустым raw-образом. Разобраться с утилитой *qemu-img* не составляет труда, мы же, чтобы заодно узнать побольше о Linux, воспользуемся утилитой *dd*.

Перед тем, как создать образ, необходимо определить его максимальный размер. Размер выбирается с учетом места, занимаемого установленной ОС, размера swap-раздела (если он используется) и объема свободного места, необходимого вам для работы. Я выбрал размер равным 3 Гб. Формат raw не поддерживает сжатие, но если создаваемый образ находится на разделе **ext2/ext3** (или другой ФС, поддерживающей «дыры» – holes), то «сжатие» будет производиться автоматически драйвером файловой системы (за счет преобразования длинных последовательностей нулей в более короткие), и размер образа будет занимать меньше, чем 3 Гб. Для того, чтобы посмотреть реальный размер, занимаемый файлом, можно выполнить команду ls -ls.

Создание нового образа выполняется командой:

```
dd if=/dev/zero of=<путь_к_файлу_образа> count=3G
```



› Рис. 1. Загружаем LiveCD Games Knoppix без отрыва от производства.



к быстрому старту

Все просто: содержимое устройства **/dev/zero** (а это одни нули) копируется в файл образа, при этом количество копируемых данных ограничено тремя гигабайтами.

Теперь, наконец, можно приступать к установке самой ОС. Для этого можно воспользоваться следующей командой:

```
qemu -cdrom /dev/cdrom -hda <путь_к_файлу_образа> -boot d
```

Опция boot позволяет изменить устройство, с которого производится загрузка. По умолчанию это **hda** (если устройства **hda** нет, то загрузка будет производиться с устройства **cdrom**), для загрузки с **cdrom** необходимо указать опцию boot с параметром d. Теперь можно загружаться в виртуальную машину и приступать к установке ОС. Созданный нами образ диска не содержит таблицы разделов, поэтому в процессе установки ОС необходимо осуществить разбиение диска. В нашем случае создание разделов осуществляется через инсталлятор Debian. Далее установка ОС производится стандартным образом.

После того, как первый этап установки будет пройден, вас попросят перезагрузиться. При этом необходимо учесть, что повторная загрузка должна выполняться уже с устройства **hda**, а не с **cdrom**, как в начале установки. Команда для повторной загрузки будет следующей:

```
qemu -cdrom /dev/cdrom -hda <путь_к_файлу_образа>
```

Как видно, исчезла опция boot, так что теперь загрузка будет производиться с устройства **hda**. После повторной загрузки можно закончить установку системы.

После того, как система будет установлена, загрузите её можно с помощью команды:

```
qemu -hda <путь_к_файлу_образа>
```

Наконец, можно приступить к настоящей работе в установленной ОС! На **рис. 2** показана загруженная в QEMU ОС Debian 3.1 с рабочим столом GNOME.



› **Рис. 2. Debian 3.1 в QEMU.**

Но это еще не все! Программа QEMU предоставляет возможность управлять своей работой из специальной консоли. Для переключения в консоль команд нужно воспользоваться комбинацией клавиш Ctrl-Alt-2. С ее помощью можно подключить новое устройство, когда ОС уже загружена (например, для подключения устройства cdrom необходимо ввести **change cdrom /dev/cdrom**), сохранить состояние виртуальной машины в файл (попробуйте **savevm /debain.state**), загрузить состояние виртуальной машины (команда loadvm) и многое другое. Для получение более подробной информации по консоли QEMU воспользуйтесь командой help.

QEMU умеет значительно больше, чем может показаться на первый взгляд – мы даже не упомянули о таких возможностях, как эмуляция сети, загрузка ядра Linux, подключение USB-устройств и о многом другом. Думаю, с этим вы справитесь и без меня, ведь главное – дать отправную точку. LXF

Почему RAW?

QEMU поддерживает большое количество форматов дисковых образов, в том числе, **cow** (User Mode Linux), **vmdk** (VMware 3/4), **cloop** (сжатый loop-файл, обычно используется различными LiveCD) и **qcow** (родной формат QEMU). Многие из них предоставляют ряд полезных функций, например, реальное сжатие данных, динамическое изменение размера, шифрование. Почему же я порекомендовал вам именно **raw**? Ответ прост как и сам формат – Linux умеет работать с ним, как с обычным дисковым разделом.

Для того, чтобы прочитать таблицу разделов файла-образа, дайте команду:

```
/sbin/fdisk -l -u <путь_к_файлу_образа>
```

Вы увидите что-то вроде:

```
...

255 heads, 63 sectors/track, 0 cylinders, total 0 sectors
Units = sectors of 1 * 512 = 512 bytes

Device               Boot    Start    End    Blocks    Id  System
<путь_к_файлу_образа>1  *     63   5542424   2771181   83  Linux
...
```

Разделы образа можно смонтировать командой mount:

```
mount -o loop,offset=32256 -t ext3 <путь_к_файлу_образа> /mnt/iso
```

Число 32256 получается путем умножения размера сектора (в нашем случае – 512 байт) на адрес начала раздела (параметр Start в выводе fdisk; в нашем случае это 63). После того, как диск будет примонтирован, вы можете получить доступ к его содержимому, при условии, что файловая система, используемая гостевой ОС, поддерживается ядром Linux хотя бы на чтение.

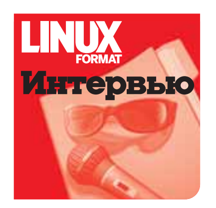


ПОНАЧАЛУ

«Впервые поглядев на Java, я подумал, что это — какая-то шутка.»

Железный человек

Ненавидите .NET? Джим Хаганин [Jim Hugunin] тоже ненавидел, пока сам не попробовал – после этого он написал IronPython. Что думает сообщество пользователей Python по поводу его неуемной жажды программирования?



Джим Хаганин любит программировать на Python – динамическом языке, про который он впервые услышал от разработчика Павла Кэртиса [Pavel Curtis]. Но ему нравится ещё и расширять его, и его любопытство насчет Java заставило его создать Jython, версию для Java-разработчиков. То же произошло и с .NET – Джим просто не мог поверить, что все настолько плохо, как говорят, потом убедился, что не настолько, и создал IronPython для использования с такими API, как .NET и Mono. Возможно, именно это любопытство и увело его от открытого кода, который поддерживал Кэртис, и Хаганин сейчас работает в Microsoft. И, как он сказал **Полу Хадсону**, не жалеет об этом.

LXF ВИЗИТКА

Джим Хаганин

» Выпускник MIT, фокусник и серийный исполнитель Python. Работал в Xerox Parc над AspectJ – расширением Java. Пришел в Microsoft в 2004, чтобы возглавить разработку IronPython и улучшить поддержку в .NET динамических языков.

Возраст	35
Национальность	Американец
Использует Linux	11 лет
Языки программирования	12
Количество ПК	4
Ежедневное потребление кофе	1 чашка

Top Trumps courtesy: Winning Moves UK Ltd

Linux Format: Как вы считаете, что это за инстинкт в вас такой, которому все время надо заново переделывать Python?

Джим Хаганин: Это сложный вопрос. До работы над Jython я трудился над проектом под названием Numeric Python. Я тогда завершал обучение в MIT, и мне приходилось делать множество научных вычислений. Раньше я пользовался языками типа Matlab, и был серьезно расстроен тем фактом, что Matlab – отличный язык для вычислений и для многого другого, что нужно ученым… но как только надо что-то спрограммировать и требуются словари, списки и классы, он просто «ломается», и мой код в итоге получался пугающе уродливым.

В MIT я возился со многими другими языками программирования: одним из них был Scheme, а еще – Perl, Python, многие языки этого класса. И мне по-настоящему понравился Python, я начал развлекаться с его возможностью расширения объектов – очень легко расширить Python новыми объектами – и вдруг обнаружил, что могу добавлять в Python все эти числовые массивы, которые были основными в Matlab. Добавляя эти числовые массивы, я чувствовал, что, программируя на Python, программирую на Matlab. Куда бы я ни метнулся – я чувствовал мощь Python. Вот так я и «подсел» на него впервые. И это была положительная сторона.

LXF: И он работает!

ДжХ: Да, он отлично работает, и сегодня Numeric Python по-прежнему довольно широко используется. Но вот расширение языка в C меня сильно расстроило. Потому что, и это проходит красной нитью через всю историю, согласно терми- »

» нологии Microsoft... «управляемый язык» (managed language) в основном означает язык, предусматривающий сборку мусора, рефлексию, безопасность типов и динамическую загрузку.

Именно этими функциями всегда и обладал Python. А вот в C их нет. В этом огромное расхождение между миром C и миром Python, так что, когда бы я ни взялся писать расширения для Python на C, всегда расстраиваюсь, потому что вдруг ни с того ни с сего приходится снова задумываться об управлении памятью или о безопасности типов. И когда я работал над этим проектом, наткнулся на Java. Впервые поглядев на Java, я подумал, что это какая-то шутка, потому что он работал ничуть не быстрее Python. При выполнении сложных вычислительных задач, например, быстрого преобразования Фурье (Fast Fourier Transform, FFT) – такие алгоритмы C обрабатывает в сотни раз быстрее, чем Python – я попробовал это делать на Java, и там эти операции проходили на той же скорости, что и в Python. И тогда я подумал: «Кому нужна смена платформы, если она не дает выигрыша по скорости?»

А потом – и это правда, хоть и кажется, что я это придумал, потому что сейчас работаю в Microsoft – именно Microsoft-версия Java убедила меня в ошибочности этого мнения. Потому что Microsoft стала первой фирмой, создавшей очень быстрый, JIT-компилятор для Java, и я стал работать с ним, и мои показатели сразу же изменились. Теперь те же серьезные и трудные вычисления на Java делались почти так же быстро, как и на C.

И я увидел все преимущества Java над C, и мне стало интересно, смогу ли я заставить Python и Java работать вместе. А окончательно убедил меня приняться за Jython Аарон Уотерс [Aaron Waters], и я очень ему обязан. Я сказал: «Круто бы вышло», а Аарон Уотерс ответил: «Да тебе нипочём не удастся заставить Python работать на JVM [Java Virtual Machine]». Я думаю, это была уловка Аарона, чтобы меня подначить – знаете, вам говорите, что нечто невозможно сделать, а это… воодушевляет.

Я уселся за работу, попробовал запрограммировать, и одним из этих сказочных проектов стал Jython. Это – один лучших проектов, над которыми я работал, потому что он становился все проще и проще, а это весьма нетипично для проектов разработки ПО. Обычно сначала идет нечто вроде медового месяца, когда все отлично работает, а потом ты вдруг осознаешь: «Боже, я ведь забыл и про то, и про это!»

Я понял, что JVM обладает всеми функциями, которые я искал, подходящими для Python и по производительности сравнимыми с C и C++. Я открыл для себя много нового о том, как работает рефлексия, и загрузчики классов, и как вообще все это работает, и постепенно проект становился все проще. Я обнаружил, как хорош Java в качестве модели расширения, поэтому с того момента мне больше не нужен был C, теперь у меня был этот замечательный мягкий переход. Когда мне нужно было сделать что-то, как в C, я мог это сделать на Java с той же самой производительностью. Вот почему я занялся Jython. Я думаю, это было чудесно.

LXF: Вы вышли из сообщества пользователей Python и занялись AspectJ. Откуда же появился IronPython?

ДжХ: IronPython начался просто для потехи. Я прочел все о новой общей языковой среде Microsoft, об этой новой классной штуке, которая появилась у Microsoft и может работать со всеми этими языками. И я над ними только смеялся, потому что я знал, что Java достаточно эффективно работает с сотнями языков. В частности, меня поразило то, что все утверждали, что с динамическими языками он работать не может. В некоторой степени это произошло из-за ранних экспериментов – попыток заставить Python и Perl работать на .NET, приведших к заключению, что это невозможно; а мне очень захотелось узнать, как это Microsoft умудрился сваять .NET настолько скверно, что он оказался даже хуже, чем JVM для работы с динамическими языками.

Я решил написать небольшую, но содержательную статью «Почему .NET – ужасная платформа для работы с динамическими языками». Это повысило бы мой рейтинг в сообществе пользователей Java, а это сообщество доставило мне бездну удовольствия и помогло мне понять структуру, когда я ею заинтересовался. Я начал этим заниматься, и вышло так же, как при разработке Jython, когда я понял, что на самом деле все работает отлично. Было и кое-что, отличающееся от начала разработки Jython: очень хороший генератор байт кода, поставлявшийся со средой времени выполнения. И производительность была очень хорошей – на 70% быстрее, чем CPython.

Так что мне кажется, я сделал это [IronPython] не из-за потребности снова и снова реформировать Python, а из желания ответить на вызов – что, дескать, это сделать невозможно…

О «БЕГСТВЕ» В MICROSOFT

«Сообщество пользователей Python сочло его реальным шансом на то, что дела IronPython пойдут хорошо.»

LXF: Так вот что сподвигло вас на это?

ДжХ: Да, а также и то, что это сработало, а потом покатилось – «Ну, это сработало, а если я еще кое-что добавлю, может, будет еще лучше?»

Так что я пошел тем же самым путем и с .NET, надо было попробовать разные способы создания многоязычной платформы. И поэтому первым моим шоком было то, что формат байт-кода в Java предполагает, что ваш исходный код написан на Java. Когда вы генерируете отладочную информацию, вы никогда не указываете, на каком языке написан этот код, потому что 99% всех людей считает, что он написан на Java. Но в байт-коде для .NET появился флажок, требующий: «Сообщите мне GUID, который вы зарегистрировали для своего языка и его реализации, потому что мы рассчитываем на использование различных языков». И снова это привело меня на тот же путь – когда отлично начинает работать то, о чем раньше говорили, что оно не может хорошо работать.

LXF: А вы не ощутили, что создание Jython и IronPython вызвало враждебность к вам со стороны сообщества пользователей Python?

ДжХ: Вы знаете, нет. Но, честно говоря, я думал, что это могло бы произойти. Я занимался Jython недолго, а потом ушел – но ушел нормально, оставив преемника, способного продолжить проект, и был уверен, что и с лицензированием, и с хостингом все в порядке, и другие смогут продолжать всем этим пользоваться и работать.

Всегда немного больно уходить из проекта, когда ты больше не можешь работать над ним, но это приходится делать, нельзя вечно быть привязанным к одному проекту. Лично я так не могу, может быть, другие и могут. Гвидо [ван Россум – Guido van Rossum, создатель Python] поступает именно так, но даже он не связан с Python слишком тесно, и он может так поступать благодаря большому пользовательскому сообществу. Вокруг Jython существует отличное сообщество, хотя и не такое большое, как сообщество пользователей CPython, поэтому и работать так же здорово у него не получается.

Однако никто не рассматривал IronPython как уход от Jython. Что мне нравится в сообществе Python – что это языковое сообщество, я думаю, наиболее открыто для различных реализаций. Они рассматривают язык – и это пошло от Гвидо – как единую крутую штуку, но реализация может быть отдельной от языка, и они действительно тепло принимают любые другие платформы. Python уже долгое время лучше всего поддерживает Windows среди «P-языков», во многом благодаря [консультанту Microsoft] Марку Хэммонду (Mark Hammond), проделавшему огромную работу, но в некоторой степени и благодаря своему сообществу, которое утверждает: «Windows – отличная вещь, мы хотим, чтобы Python хорошо работал под Windows. А еще мы хотим, чтобы он отлично работал на Linux и Mac...»

Jython рассматривали как еще одну возможность для Python глубже проникнуть в ряды разработчиков Java. А что касается IronPython – похоже, его рассматривают так: «Вот отличная возможность продвинуть Python в новое крутое пространство .NET». Я не чувствую, что разработчики Jython и IronPython предаются скорби.



› «Список проколов длинный, Джим.» Ответы нашего героя читайте на www.linuxformat.co.uk/hugunin.html.



Не один Джим связал свою жизнь с крупной корпорацией – создатель Python, Гвидо ван Россум, работает в Google.

Кто-то хотел бы моего возвращения – чтобы я все свое время посвящал Jython, но есть и такие, кто хочет, чтобы я много чем занимался.

LXF: А спрашивали: «Почему вы пошли в Microsoft?»

ДжХ: Ну, не особо. Я представил IronPython сообществу Python в марте 2004 в качестве независимого консультанта, планирующего, чем заняться дальше. А затем, через четыре-пять месяцев, я объявил на OSCon, что ухожу в Microsoft. Люди просто видели IronPython, и знать не знали, что я ушел в Microsoft, они раздумывали: «А будет ли IronPython летать?». Они понимали, что это – крутая штука, но были не совсем уверены, что это произойдет. Я тратил свое время, занимаясь консалтингом – жить-то надо, и большая часть моего времени уходила на это. Так что, я думаю, мой уход в Microsoft был воспринят как реальный шанс на процветание проекта IronPython.

Интересно, как все сложилось бы, поступи я по-другому. Если бы мое первое заявление об IronPython звучало, как «Я ухожу в Microsoft», и люди не успели бы привыкнуть к одному до того, как узнали о другом, вряд ли реакция была бы такой, как сейчас.

LXF: При работе над IronPython в MS вы прошли через GotDotNet [www.gotdotnet.com] – схему частично открытого кода (Shared Source).

ДжХ: Да.

LXF: И что вы думаете по этому поводу, есть ли разница – до и после Microsoft?

ДжХ: Не собираюсь сводить какие-либо счеты по этому поводу: GotDotNet не является моим любимым сайтом частично открытого кода (Shared Source). Я не считаю, что он – из той же лиги, что большинство сайтов разработки программ с открытым/частично открытым кодом. Я полагаю, что он просто нужен Microsoft, чтобы лучше работать в нем. Но Microsoft учится играть в этом пространстве. Это никогда не было основным бизнесом Microsoft, но Microsoft делает все больше и больше – мы создали WTL [Windows Template Library] и Flex Wiki [открытые проекты Microsoft, размещенные на SourceForge, – прим. ред.].

Мы продолжаем выяснять, насколько эффективно Microsoft может играть в этой области без тотальных изменений в своей основной деятельности. GotDotNet – пример того, что впереди очень долгий путь, по моему личному мнению.

LXF: Но, по крайней мере, они уже в пути!

ДжХ: Тот факт, что GotDotNet существует, уже здорово. Это – большой шаг в верном направлении. Но я надеюсь, что мы зайдем намного дальше. Конечно, мы сменили направление… у нас отличная обратная связь с нашими пользователями, многие из них не в восторге от GotDotNet. В списке рассылки полно сообщений с жалобами по этому поводу. А мы ответили, может быть, лишь на половину из жалоб. Мы закрыли форумы, но продолжаем поддерживать списки рассылки, чтобы сохранить возможность эффективной дискуссии.

Люди хотят, чтобы у нас был репозитарий открытого кода. Но на данный момент у нас просто нет инфраструктуры, которая могла бы рационально и разумно выполнить эту работу, поэтому мы делаем релизы каждые две недели. Конечно, это не то же самое, что иметь древо с открытым исходным кодом, но мы стараемся, чтобы у пользователей сохранялось ощущение интерактивности – если вы что-то предложили, то через неделю это, вероятно, уже будет проверено и вы получите ответ. Чтобы узнать, как заставить многие программы .NET работать эффективно, нам нужна обратная связь от пользователей про то, что работает на этих программах, и обратная связь не должна прерываться.

LXF: Интересно, какова оборотная сторона лицензии Shared Source License? Вы говорите, что она короткая и ясная. А есть ли у нее недостатки, с точки зрения открытого кода?

ДжХ: Тот факт, что она называется Частично Открытым кодом – я думаю, это неудачно. Это скорее восприятие лицензии, а не реальность. Я не думаю, что с лицензией что-то не так. Это – лицензия в стиле BSD, я сам их предпочитаю, но вы можете применить ее к коммерческому коду, и вы не обязаны открывать свой коммерческий код всему миру, так что получается, что терминология лицензии мне очень нравится.

LXF: Значит, она похожа на LGPL?

ДжХ: Больше всего она похожа на Лицензию BSD. А самой похожей, наиболее близкой, наверное, будет MPL [Mozilla Public License].

LXF: Что вы думаете о работе IronPython на Mono?

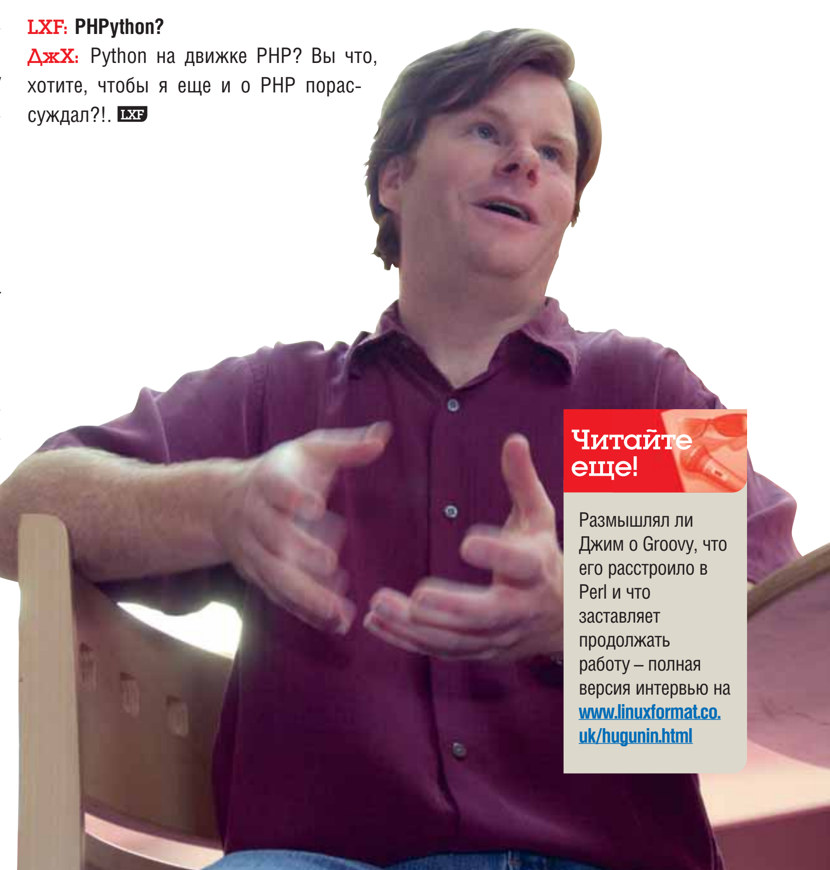
ДжХ: Для начала – предисловие: все, что я вам сказал – мое личное мнение. А то, что я вам скажу, тем более мое личное мнение. Когда я заговорил об IronPython, я прежде всего сказал, что есть отличная реализация .NET. Она отличная по двум причинам. Одна – те огромные усилия, которые Microsoft в нее вкладывает, и я вижу колоссальный импульс, исходящий оттуда, и она станет очень эффективной, успешной и мощной. Вторая причина в том, что это – международный стандарт, стандарт ECMA/ISO, и это действительно стандарт, потому что есть люди, которые создают его альтернативные реализации, а не выдумывают свой. Так что для меня Mono – это частичное объяснение причины, по которой я, как разработчик языков, нахожу .NET особенно интересным. Это платформа, и она широка. В конце моего первого выступления на OSCon 2004, IronPython работал только на моем ноутбуке. Единственным компьютером в мире, на котором он работал, был мой ноутбук с Windows. К концу моего выступления его уже скачали, и он начал работать на OS X и Linux – причем эффективно работать. Мне это показалось потрясающим. А парни из Mono делают отличную работу. Производительность улучшается от релиза к релизу, и мне нравится наблюдать за этим. Здесь я это вижу.

LXF: А каково ваше будущее в Microsoft? Вы будете продолжать работу над IronPython или займетесь, ну, не знаю, SchemePython?

ДжХ: Нет, нет, лучше PhyrricPython, ладно?

LXF: PHPython?

ДжХ: Python на движке PHP? Вы что, хотите, чтобы я еще и о PHP рассуждал?!. LXF



Читайте еще!

Размышлял ли Джим о Groovy, что его расстроило в Perl и что заставляет продолжать работу – полная версия интервью на www.linuxformat.co.uk/hugunin.html

Новая серия! Web-сервер Apache для начинающих web-мастеров и администраторов



ЧАСТЬ 1: По данным www.netcraft.com, более 60% web-серверов по всему миру работают под управлением *Apache*. Причиной такой популярности является непревзойдённая гибкость и мощь этого сервера наряду с высокой скоростью работы. **Сергей Супрунов** поможет вам принять его на службу...

Немного истории

В начале 90-х годов прошлого века основным сервером только-только зарождающейся Всемирной паутины был *httpd*, в разработке которого активное участие принимал Национальный Центр Суперкомпьютерных Приложений (NCSA). Однако, когда в 1994 году проект покинул один из ведущих разработчиков, работы были свернуты. К счастью, NCSA был правительственной организацией, и он не старался заполучить себе все права, так что исходный код *httpd* был доступен всем желающим. Благодаря этому даже после официального закрытия проекта люди продолжали писать к серверу исправления, дополнения и т.д.

И вот в начале 1995 года Брайан Белендорф (*см. LXF80*) и Клиф Скольник, взяв за основу последнюю версию *httpd*, выпущенную NCSA, и применив к ней огромное количество найденных в сети патчей, выпустили первую версию того, что ныне известно под именем *Apache* [Собственно, название происходит от «A patchy server» – пропатченный сервер, – прим.ред.]. С тех пор сервер активно развивается, и сейчас является одним из самых успешных открытых проектов.

Установка

В настоящее время идет разработка трех ветвей сервера – *Apache 1.x* (текущая версия – 1.3.37) и две ветки *Apache 2.x* (последние версии - 2.0.59 и 2.2.3 соответственно). Конкретные дистрибутивы могут содержать более ранние версии.

Если *Apache* включен в состав вашего дистрибутива или доступен для установки из репозитория, предпочтительно использовать эту возможность. Впрочем, при необходимости инсталляцию можно выполнить и из исходных кодов. В данной статье я использовал версию *2.0.55*, установленную в Ubuntu с помощью Synaptic. Про инсталляцию из исходных текстов рассказано во врезке «Ручная работа».

Настройка

К сожалению, разработчики дистрибутивов пока не пришли к единому мнению на счет того, как же лучше именовать те или иные файлы. В «чистом» пакете, скачанном с **Apache.org**, предполагается, что основной двоичный файл именуется **httpd**, главный файл конфигурации – **httpd.conf**, а утилита управления - *Apachectl*. То есть так, как это принято и в версиях 1.x.

Однако в некоторых дистрибутивах применяют несколько иные схемы. Например, Ubuntu, Knoppix (да, похоже, и все семейство Debian) размещают конфигурацию в **/etc/Apache2**, а главный файл конфигурации называется **Apache2.conf**. Файл **httpd.conf** тоже присутствует, но используется в основном для обратной совместимости. А, скажем, Fedora Core придерживается на сей счет более «официальной» точки зрения, и размещает основной файл настройки в **/etc/httpd/conf/httpd.conf**, с той лишь разницей, что часть вспомогательных настроек вынесена в отдельные файлы в каталоге **conf.d**.

В принципе, вариантов не так уж и много, и свой конфигурационный файл, думаю, вы отыщете без труда. В дальнейшем я буду ссылаться на него как на **Apache2.conf**, так что не забывайте вносить соответствующие поправки...

Итак, заглянув в **Apache2.conf**, вы увидите там несколько видов строк:

» начинающиеся с символа «#» - это комментарии, они игнорируются при анализе файла и служат лишь для пояснения нижележащих опций;

» начинающиеся с «<» - это синтаксические конструкции, разделяющие конфигурационный файл на так называемые секции;

» обычные директивы вида «имя значение».

Секции тоже можно условно разделить на два типа: «ограничители» и «условия». Первые ограничивают область действия заключенных в них директив. Например, секция <Directory /var/www/special> будет влиять только на запросы файлов, размещенных в указанном каталоге. Секция <Location /server-status> будет отвечать за запросы ресурса с URL вида http://yourserver.ru/server-status/... Секция <Limit GET> окажет влияние только на GET-запросы. К этому же виду секций можно отнести и виртуальные хосты, с которыми мы познакомимся чуть позже.

Секции-условия позволяют применять те или иные директивы только в том случае, если выполняется некоторое условие, например, если подключен конкретный модуль: <IfModule prefork.c>. Причем влияют эти директивы на конфигурацию всего сервера (или той секции-ограничителя, в которую вложена данная секция-условие).

Директивы отвечают за самые различные аспекты поведения web-сервера – от прослушиваемых портов до подключения модулей и обработчиков (handlers). Обращайте особое внимание на директивы Include, которые позволяют вставлять в текущий файл настройки строки из других файлов (особенно этим «грешит» Ubuntu). Другие важные для нас директивы будут рассмотрены по ходу дальнейшего изложения.

Первая страничка

Перейдем к практике. *Apache* мы поставили и даже можем увидеть его тестовую страницу, если введем в браузере http://yourdomain.ru (если вы ставите сервер на той же машине, на которой работаете, то можно ввести http://localhost). Опять-таки, в зависимости от дистрибутива, это может быть либо страничка с «перышком» (см. рисунок), либо оглавление каталога (как в Ubuntu), либо просто строчка «It works!» (как в официальном пакете). Давайте посмотрим, откуда она берется.

В конфигурационном файле отыщите строку DocumentRoot - она укажет вам на каталог, который *Apache* считает корневым для сайта. В зависимости от дистрибутива и способа инсталляции это может быть **/var/www**, **$prefix/htdocs** (где **$prefix** - каталог, куда выполнялась установка), **/var/www/html**. В нем вы должны найти файл с именем **index.html**. Если вы вернетесь к конфигурации и найдете строку DirectoryIndex, то убедитесь, что файл с таким именем и должен возвращаться на запрос, в котором конкретное имя ресурса не указано. Таким образом,

Ручная работа

Менеджер пакетов, это, конечно, хорошо. Но какой же настоящий линуксоид не попробует хотя бы раз в жизни поставить что-то из исходников? Благо, и повод у нас для этого весьма благородный – хочется 2.2.3, а в репозитарии есть только 2.0.55. Так что – в путь!

1. Скачайте архив с последней версией с сайта http://httpd.apache.org:

```
wget http://apache.rinet.ru/dist/httpd/httpd-2.2.3.tar.bz2
```

2. Распакуйте его в свой домашний каталог:

```
tar xvjf httpd-2.2.3.tar.bz2
```

3. Сконфигурируйте исходный код:

```
cd httpd-2.2.3
./configure --prefix=/usr/local/apache22
```

С помощью --prefix мы изменили каталог установки, чтобы случайно не затереть уже установленный 2.0.55. С помощью опций enable/disable вы можете изменить «умолчательный» состав модулей, которые будут собраны в основном двоичном файле.

4. Наконец, традиционная сборка и инсталляция:

```
make
sudo make install #
```

или

```
su -c 'make install'
```

в дистрибутивах, отличных от Ubuntu

друг «бледнолицых»

чтобы вместо этой открывался ваш html-файл, просто замените им индексный (главное, чтобы он был доступен для чтения пользователю www-data, с правами которого обычно работает *Apache*). Впрочем, никто не запретит вам указать здесь свой каталог.

Автоиндекс

А что *Apache* будет делать, если в каталоге отсутствует индексный файл? Это также зависит от настроек. По умолчанию сервер собирается с поддержкой модуля mod_autoindex (для проверки выполните команду **Apache2 -l** или **httpd -l**, в зависимости от имени исполняемого файла в вашем дистрибутиве), так что *Apache*, используя его возможности, автоматически построит страничку-оглавление со ссылками на все файлы, содержащиеся в запрошенном каталоге. С помощью директивы IndexOptions можно подстраивать вид этой страницы.

Данная функция удобна для создания каталога файлов или чего-то еще, организованного по принципу FTP-сервера. Но если вы считаете такую возможность излишне опасной (не всегда и не всем нужно показывать, что и где у вас лежит), то этот модуль можно отключить:

```
Options -Indexes
```

Кстати говоря, директива Options очень важна и для других аспектов безопасности. Обязательно ознакомьтесь с ней подробнее в документации. После изменения конфигурации не забудьте перезагрузить сервер: *Apachectl restart* (в некоторых дистрибутивах эта утилита называется *Apache2ctl*).

Домашние страницы

Вы, имея привилегии root в системе, можете без ограничений править свою интернет-страничку. Но что, если вы хотите предоставить такую возможность своему другу, который зарегистрирован в вашей системе, но без прав суперпользователя? Не давать же ему неограниченный доступ?

Для решения этой задачи предназначен модуль mod_userdir. Если он подключен, то вы можете указать в конфигурации такой фрагмент (скорее всего, он там уже есть, но закомментирован):

```
UserDir public_html
<Directory /home/*/public_html>
  Options Indexes
  AllowOverride Indexes
</Directory>
```

Здесь мы указываем, что при получении запроса вида http://yourserver.ru/~vasya нужно открывать индексный файл из домашнего каталога пользователя vasya, т.е. из **/home/vasya/public_html**. Ну а там Вася пусть делает все, что ему захочется. Чтобы он не сильно шалил, его возможности можно ограничить в последующей секции <Directory> для каталогов, соответствующих указанной маске (в примере мы устанавливаем ему опцию Indexes и позволяем ее переопределять самому). Кстати, о переопределении конфигурации...

Файл .htaccess

Конфигурационный файл позволяет настроить любые опции для любого каталога. Но здесь есть два неудобства – во-первых, для его редактирования нужны права суперпользователя, и во-вторых, после каждого изменения нужно перегружать сервер (для чего, опять-таки, нужно обладать правами root).

Apache позволяет управлять параметрами доступа к каталогам более гибко - путем размещения в них специальных файлов, обычно с именем .htaccess. Если для данного каталога действует опция AllowOverride All, то в .htaccess-файле можно указывать практически любые опции, допустимые для секции <Directory>. Однако администратор может ограничить эти возможности, задав более жесткие ограничения, как в примере выше.

Учтите, что использование **.htaccess** существенно повышает нагрузку на сервер – для каждого запроса серверу придется проверять наличие этого файла в каждом каталоге от корня файловой системы до месторасположения запрошенного файла, так что если в них нет острой необходимости, лучше пользуйтесь общим файлом конфигурации. Полностью запретить эти файлы можно с помощью директивы AllowOverride None.

Виртуальные хосты

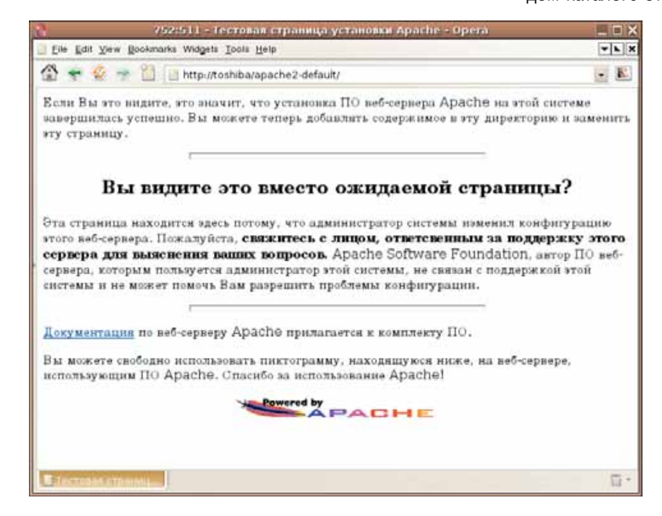
Домашние странички пользователей достаточно удобны для задач «домашнего» уровня. Но *Apache* предоставляет более мощное средство размещения нескольких сайтов на одном сервере (и под одним IP-адресом) - виртуальный хостинг. Рассмотрим простейший пример:

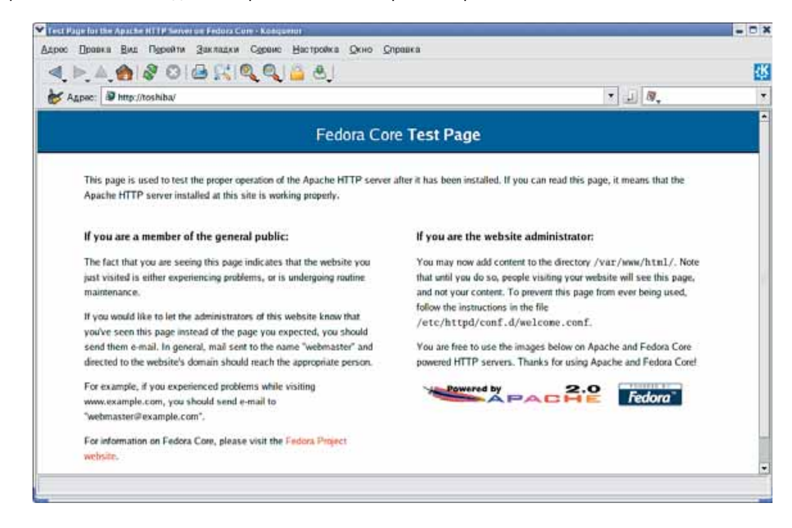
```
NameVirtualHost 127.0.0.1
<VirtualHost localhost>
  DocumentRoot /var/www/localhost
</VirtualHost>
<VirtualHost toshiba>
  DocumentRoot /var/www/toshiba
</VirtualHost>
```

Здесь мы указали, что у нас будут виртуальные хосты по адресу 127.0.0.1, и дальше перечисляем секции <VirtualHost> для каждого доменного имени. В этих секциях обязательно нужно указать корень сайта (DocumentRoot) и, при желании, любые другие настройки. Разместив в указанных каталогах индексные файлы с различным содержанием, вы получите два сайта, открывающиеся в зависимости от того, какое доменное имя будет указано в запросе.

Чтобы виртуальный хостинг работал правильно, все доменные имена должны быть описаны на сервере DNS (для тестирования будет достаточно файла **/etc/hosts**), а браузер клиента должен поддерживать протокол HTTP/1.1, так как HTTP/1.0 не поддерживает поле Host: заголовка, в котором и передается имя домена. А поскольку запрос направляется по IP-адресу, то для *Apache* это поле - единственный способ узнать, какой же из сайтов запрашивается клиентом.

Изложенных сегодня сведений должно быть достаточно, чтобы начать работать с *Apache* и создавать несложные статические странички. Главное - не бойтесь экспериментировать! **LXF**







Matt Herring

Что сделала для нас OSDL?

Кажется, что The Open Source Development Labs (Лаборатория Разработки Открытого Кода) сидит на нескольких стульях – но знает ли кто-нибудь, чем конкретно она занимается? Расследование ведет **Грэм Моррисон**.

Прочтя интервью со Стюартом Козном [Stuart Cohen] пару номеров назад, вы, наверное, захотели узнать побольше о его нанимателе. Козн – исполнительный директор OSDL, организации, образованной в 2000 г. консорциумом больших компаний, в том числе, CA, Hitachi, HP, IBM, Intel и NEC. Эти фирмы предоставили финансирование и получили центр разработки Open Source, беспристрастный и некоммерческий, который посвящен продвижению Linux (на что многие из них делают ставку) в большой бизнес. Это мир подготовительных комитетов и спецификаций, рабочих групп и официальных документов: частично – мозговой центр, частично – глобальный консорциум, и частично – Международное Спасение.

«OSDL» звучит несколько отчужденно, и тем не менее лаборатория сильно влияет на большое сообщество Linux – не только потому, что платит зарплату Линусу Торвальдсу, но и потому, что эта организация позиционировала себя как «центр тяжести» (термин, от которого она теперь дистанцируется) Linux, в который мы все вносим свою лепту. OSDL открыта для всех, а не только для многонациональных конгломератов. Отдельные личности могут свободно присоединиться и принять участие во всевозможных рабочих группах.

За лабораторией стоят хорошо продуманные и хорошо финансируемые проекты. Например, она сформировала фонд юридической защиты для помощи в борьбе пользователей Linux против угроз возбуждения судебных дел, ее рабочие группы разрабатывают исключительно подробные спецификации; ею нанято по меньшей мере 10 разработчиков из тех, что напрямую влияют на функции Linux-систем, используемых нами ежедневно – половина годового бюджета лаборатории, а это $5 млн., тратится на развитие технологий.

Но от большого бизнеса не уйдешь. Любое дело OSDL, будь то тестовая лаборатория или сервисы технической поддержки, движимо желанием вывести Linux на уровень предприятий. Фирмы-учредители задались направлением на совершенствование преимуществ открытого кода. Это означает, что они могут управлять политикой разработки и предлагать области исследований – если в этом заинтересован каждый член. OSDL не отстраняет от дела ни своих членов, ни другие открытые инициативы.

Хотя OSDL и «открытая», но все же оставляет ощущение некоторой загадочности. Поэтому мы собираемся присмотреться к некоторым сотрудникам OSDL и крупнейшим ее проектам и решить, что же служит стимулом для этой организации. Для сначала сразу предупредим: поскольку это большой бизнес, то будет много акронимов. Мы пытаемся поменьше употреблять их, но все же их будет много. С другой стороны, мы постарались свести к функциональному минимуму использование слова «рычаг», так что можете читать без опасения запутаться.

Хакеры высокого полета

Вот реальные причины важности OSDL.

Одной из причин известности OSDL является то, что в ней работают высококлассные хакеры ядра. Отчасти это потому, что она любит фанфары, а отчасти потому, что никто не сможет игнорировать организацию, трудоустроившую Торвальдса. Альтруистический аспект мотивации найма на работу Линуса Торвальдса, Эндрю Мортона и иже с ними в том, что OSDL хотела предоставить им условия разработки без конфликтов с другими обязанностями, чтобы эти два разработчика смогли целиком сосредоточиться на разработке ядра Linux. Но нельзя отрицать, что OSDL получает свою долю аплодисментов, когда люди видят ее логотип на их визитных карточках.

Нет ни одного пресс-релиза от OSDL, который бы не включал следующую сноску: «OSDL – спонсор Линуса Торвальдса, создателя ядра Linux – призвана ускорить рост и адаптацию операционных систем на основе Linux на предприятиях». Без сомнения, немалой частью оказываемого ей доверия OSDL обязана найму Линуса Торвальдса, человека, с которого все началось. До перехода в OSDL в июне 2003 г., Торвальдс был сотрудником корпорации Transmeta (члена OSDL), разрешавшей ему в рабочее время повозиться с ядром Linux, помимо разработки ее собственного ПО. Переход позволил Торвальдсу сосредоточиться исключительно на основных компонентах ядра Linux.

Итак, Линус впервые стал официально получать зарплату за сопровождение ядра Linux и координацию процесса разработки. Стюарт Коэн сказал тогда: «Принятие Линусом предложения присоединиться к нам подтверждает важность нашей миссии. OSDL – единственная организация, в которой разработчики Linux, потребители и дистрибьюторы могут участвовать на равных. Дополнение лаборатории взглядами и руководством Линуса увеличивает нашу значимость для всех трех групп».

Эндрю Мортон присоединился к Линусу в OSDL через месяц, в июле 2003 г. Мортон – «хранитель» ядра 2.6 (Торвальдс работает над текущей версией ядра), то есть лицо, ответственное за включение заплаток от третьих лиц в древо ядра текущей версии. Недавно, он, к недоумению многих, предложил посвятить следующий цикл разработки ядра исправлению ошибок, закравшихся в текущую версию. Было очень ценно, что два основных разработчика Linux работают в одной компании, причем бок о бок. Но в августе Мортон ушел в Google. Это может нарушить нейтральность, которой разработчики наслаждались в OSDL, где отсутствовало прямое давление со стороны крупных фирм, являющихся частью OSDL.

Теплица для ядра

OSDL получила своего второго сотрудника в январе 2005 г., когда ее ряды пополнил Эндрю Триджелл, ведущий разработчик *Samba* – одного из самых важных существующих проектов с открытым кодом. Без *Samba* мы не могли бы совместно использовать любые файлы или принтеры с Windows-машинами в локальной сети, а это необходимое звено при развертывании Linux в любом офисе или на предприятии. OSDL позволила Триджеллу работать над ней полный рабочий день, но, завершив запланированные работы над четвертой версией комплекса приложений *Samba*, он ее покинул и вернулся на свою предыдущую должность в IBM, оставаясь лидером проекта *Samba*. Уходя, Триджелл выразил благодарность OSDL за ее участие в проекте: «Обладать временем, позволяющим сосредоточиться только на одном проекте, действительно здорово. Большое спасибо OSDL за участие и поддержку разработки *Samba*».

»



› Трудоустройство Линуса Торвальдса дало OSDL рекламную раскрутку.



› Эндрю Мортон все время пребывания в OSDL занимался ядром.



› Тридж ушел, как только закончил работу над Samba 4.

Проект Open Driver



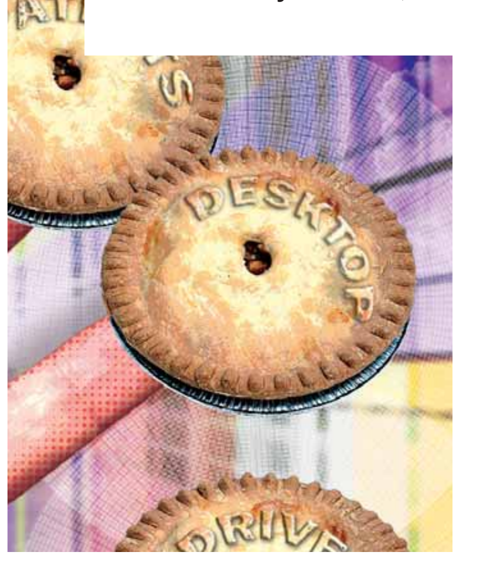
В поисках новых альтернатив, OSDL часто затевает и финансирует проекты. Наиболее масштабные ее кампании выросли из поддержки открытого форума San Francisco Linux World Expo в августе 2005. Форум был посвящен обсуждению препятствий разработке драйверов с открытым кодом, и на нем присутствовало много разработчиков драйверов ядра. Форум обозначил потребность в централизации разработки драйверов – явно в унисон с философией «центра притяжения» OSDL, и лаборатория взяла на себя создание web-сайта, способного играть роль центра разработки драйверов.

В результате возник проект Open Driver, с домашней страницей http://developer.osdl.org/dev/opendrivers. Сайт содержит новейшую информацию, необходимую для создания и установки ваших собственных драйверов. Число категорий драйверов стало заметно больше изначальных пяти (носители, сеть, принтеры, видео и звук), охватив большинство других настольных устройств; среди них – сканеры, MP3-плейеры и цифровые камеры.

Оригинального контента на сайте не слишком много. Вместо этого, каждая категория предоставляет список самых свежих сетевых ресурсов по вашему оборудованию. Цель сайта в том, чтобы разработчики драйверов, начинающие свой путь в Linux и Open Source, имели центральный репозиторий и понимание, с чего начать. Проект Open Driver содержит свои собственные wiki и списки рассылки, а также учебный семинар, в этом году совпавший с OSCon в Портленде, штат Орегон.

Устанавливая стандарты

Если для внедрения Linux на предприятии требуется поработать с бумагами, то OSDL как раз и может здесь помочь.



Как и следовало ожидать, OSDL заинтересована в коммерческом успехе проектов, которые она финансирует и в которых участвует. В частности, за последние несколько лет созданы три рабочих группы, где представители фирм-членов OSDL встречаются для проработки спецификаций и применяют свои технические знания для решения наболевшего вопроса об ускорении внедрения Linux в трех областях.

Наверху – рабочая группа Carrier Grade Linux (CGL), проталкивающая Linux в бурно разрастающийся телекоммуникационный бизнес. Посередине – группа Data Center Linux (DCL). Ее любимое слово – «рычаги», и она старается открыть для Linux рынок вычислительных центров крупных корпораций. И, наконец, группа Desktop Linux (DTL) сосредоточена на том, что понятно нам всем, а именно на продвижении Linux в настольные среды, типичные для офисов.

Для телекоммуникаций

Внедрение в области, традиционно использующие проприетарные решения – одна из главных задач OSDL, и есть одна область, в которой можно особенно поживиться. Это индустрия телекоммуникаций. Здесь традиционно вкладываются средства в проприетарные оборудование и программное обеспечение, и безопасная, стабильная и расширяемая Linux-альтернатива принесла бы немало выгоды. Решение от OSDL известно как Carrier Grade Linux (CGL – отказоустойчивый Linux): это спецификации системы, надежной более чем на 99.999% – всего пять минут на отказ в год. Но некоторые установки требуют даже большей надежности, около 99.9999%, то есть не более 30 секунд на отказ в год. Даже при знаменитой безопасности Linux необходимо многое изменить, чтобы гарантировать такое качество.

В январе 2002 г. OSDL сформировала рабочую группу CGL (для коллекционеров акронимов – CGL-WG), чтобы создать для Linux комитет по телекоммуникациям. Основная задача группы – утверждение стандарта, который обеспечит доступную и высокоэффективную операционную систему Linux с прекрасной поддержкой, способную напрямую конкурировать с готовыми и заказными проприетарными решениями, получающими все большее распространение. Участие в ней приняли 24 организации – члены OSDL, а учредительские обязанности легли на плечи IBM, MontaVista (известной благодаря своей real-time версии ядра Linux), Intel и NTT Network Service Systems Labs, возглавляющей подкомитет разработки. К настоящему моменту имеется пять версий спецификации. Версия 3.1 вышла в июне 2005. Заговорщики из OSDL работают над версией 3.2 и намерены выпустить ее в этом году.

В спецификации CGL-WG девять основных положений, среди которых:

1 Соответствие стандартам

Это действительно залог нормальной работы CGL-WG с другими стандартами Linux и индустрии телекоммуникаций. Сюда включаются Linux Standard Base (LSB), POSIX, IPv6 и PCI Industrial Computer Manufacturer's Group.

2 Платформа

Любая CGL-система накладывает специфические требования на оборудование. Сюда входит «горячая» замена, удаленная загрузка, бездисковое выполнение приложений через сеть и архитектура Advanced Telecom Computing Architecture [Продвинутая архитектура телекоммуникационных вычислений – серия спецификаций промышленных стандартов, разработанных для следующего поколения отказоустойчивого оборудования и ПО].

3 Работоспособность

Поддержка системы в рабочем состоянии, чтобы все ее сервисы были всегда доступны, очень важна, и конкретным поставщикам приходится вносить множество изменений в ядро Linux. MontaVista – хороший пример дистрибьютора, самостоятельно переделавшего ядро Linux для удовлетворения повышенным требованиям.

4 Удобство обслуживания

Машина, работающая постоянно, должна иметь возможность сервисного обслуживания, а значит, и поставлять в процессе работы материалы для диагностики, в форме дампов активного ядра или динамической отладочной информации.

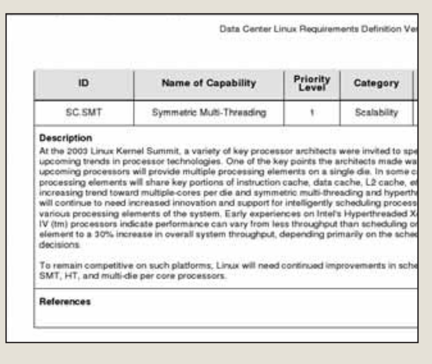
5 Кластеризация

Запуск приложений на кластерах часто является единственным способом получения высоких уровней доступности, требуемых CGL. Кластер обеспечивает повышенную производительность и резервирование использованием параллельных ресурсов. Кластеризация настолько важна, что породила свой собственный проект OSDL.

Остальные четыре положения спецификации CGL касаются производительности, безопасности, расширяемости и создания инструментов поддержки приложений. HP сделала хороший шаг в начале этого года, зарегистрировав Debian Sarge под CGL версии 2.02, что привело к появлению внутри Debian подпроекта по созданию отказоустойчивого дистрибутива Linux, удовлетворяющего спецификации CGL.

Почему важны спецификации

Для корпорации среднего размера любые инвестиции в новую технологию связаны с удовлетворением большому списку требований. Необходимо проверить спецификации любого будущего оборудования, а каждый производитель устройств будет предоставлять свои собственные спецификации. Вот тут и появляется OSDL. Она избавляет дистрибьюторов Linux от проблем создания своих собственных спецификаций, передавая все полученные из опыта знания в комитет, занятый формированием спецификаций. И все это делается открыто – любой может загрузить последнюю версию каждой спецификации, прочитать протоколы совещаний и списки рассылки рабочих групп, предоставляемые OSDL.


Data Center Linux Requirements Definition Ve

ID	Name of Capability	Priority Level	Category
SC.SMT	Symmetric Multi-Threading	1	Scalability

Description

At the 2003 Linux Kernel Summit, a variety of key processor architects were invited to spe upcoming trends in processor technologies. One of the key points the architects made wa upcoming processors will provide multiple processing elements on a single die. In some c processing elements will share key portions of instruction cache, data cache, L2 cache, e increasing trend toward multiple-cores per die and symmetric multi-threading and hyperth will continue to need increased innovation and support for intelligently scheduling process various processing elements of the system. Early experiences on Intel's Hyperthreaded X IV (tm) processors indicate performance can vary from less throughput than scheduling o element to a 30% increase in overall system throughput, depending primarily on the sche decisions.

To remain competitive on such platforms, Linux will need continued improvements in sche SMT, HT, and multi-die per core processors.

References

› После учета всех требований OSDL к ядру Linux обновились и спецификации совместимости DCL.



OSDL хочет выглядеть скорее сторонником Linux корпоративного уровня, уровня вычислительных центров, а не проталкивать открытый код на машины меньшего размера или даже на настольные компьютеры. Частично это связано с техническими аспектами CGL, а также с тем, что OSDL содержит и управляет рабочую группу DCL, цель которой – установка Linux в хранилищах данных с водяным и воздушным охлаждением некоторых крупных корпораций, многие из которых, так уж вышло, являются членами OSDL, но есть и фирмы вроде Siemens AG и Unisys.

В базе данных – пингвины?!

Поскольку CGL-спецификация очень тщательна, рабочая группа DCL может использовать ее в качестве стартовой точки для высококлассного центра обработки данных. Вычислительные центры, объединяющие корпоративные сервера приложений, базы данных и т.д., лежат в сфере ИТ с высокими стандартами. Цели группы DCL разбиты на три области деятельности, обозначенные как Передний край (Edge), Приложения, База данных. Группа определяет «Край» как сервер, где наряду с web-сервисами имеются edge-приложения – в основном ради улучшения производительности путем использования промежуточного сервера для передачи запросов к сервисам, запущенным на других машинах. Приложения и База данных в объяснениях не нуждаются.

Плодом работы группы является версия 1.2 документа, озаглавленного «Цели и Возможности DCL» и впервые выпущенного еще в 2004. Это огромный документ с добавками от Sun, HP, IBM и Intel. Он похож скорее на техническую спецификацию, в нем приводится список более 300 возможностей, которые, как считает группа, важны для принятия Linux предприятиями. Например, приоритет номер один – возможность расширения до 16 CPU (пожалуй, уже маловато), асинхронный сетевой ввод/вывод и «горячая» замена процессоров, памяти, а также устройств PCI/PCI-X.

На рабочем столе

Не удивительно, что самая молодая рабочая группа OSDL нацелена на настольный Linux. OSDL, видимо, понадеялась, что образование группы – в начале 2003 г. – будет ответом на критику заботы о внедрении Linux только на корпоративный рынок. И верно, образование рабочей группы DTL дало всему сообществу Linux возможность включиться в то, в чем оно имеет непосредственный опыт, и OSDL признает, что потенциальный размах этого проекта может затмить обе другие ее инициативы. Но группа отнюдь не чурается предприятий, и Desktop Linux включает области, которые вы не сразу отнесете к открытому рабочему столу – например, киоски и рабочие станции с фиксированным набором функций.

Вы будете разочарованы, если ожидали больше действий и меньше создания спецификаций – количество бумажной работы остается обильным. DTL формулирует свою миссию как «работа с сообществом Open Source с целью выявления большого набора моделей настольного Linux, разработки спецификации и предоставлению рекомендаций.» Вы можете найти спецификацию версии 1.0 на web-сайте OSDL, и в отличие от спецификаций CGL и DCL, она удобочитаема!

В настоящее время DTL работает при содействии представителей AMD, Novell, Intel и IBM. Одной из их наиболее интересных работ является не код, а результаты исследования, проведенного в октябре 2005 г., чтобы определить фокус действий рабочей группы. ИТ-профессионалам, внедрившим или думающим о внедрении настольного Linux в своих учреждениях, были заданы вопросы об использовании Linux. Было опрошено 3300 респондентов, и результаты удивили многих. Самым неожиданным оказался ответ на вопрос о причине выбора Linux ИТ-профессионалами. Это не экономия денег и не повышенная безопасность, как предсказывали многие эксперты, а чей-то совет. Люди использовали Linux и были им так довольны, что рекомендовали коллегам. Электронная почта, офисные инструменты и web-браузеры были признаны критически важными приложениями, а дистрибутив Ubuntu оказался самым популярным. Естественно, что результаты опроса (см. www.osdl.org/dtl/DTL_Survey_Report_Nov2005.pdf) вызвали много разговоров в сообществе Linux.

Молчаливый партнер

Критиковать OSDL за разработку одной спецификации за другой без написания хотя бы строки кода легко, но как раз это и необходимо корпоративному рынку, на который OSDL пытается пробиться. Что нам нравится меньше, так это отсутствие диалога с большим сообществом Linux – обычные пользователи Linux остаются за бортом любых разработок, выполняемых для соответствия большому бизнесу. Хороший пример – написание данной статьи, потому как даже нам было трудно получить комментарии или отзывы от людей из OSDL. А каковы шансы среднего раздраженного пользователя Linux?

Если бы OSDL не твердила столь часто, что она является «центром тяжести» разработки Linux, мы были бы счастливы причислить OSDL к компаниям вроде CollabNet, продвигающей открытые стандарты и изменяющей способ ведения бизнеса крупными предприятиями. Но это не так. OSDL финансируется несколькими крупнейшими фирмами ИТ-бизнеса и наняла двух самых влиятельных из имеющихся разработчиков Linux. Если OSDL желает рассеять дымовую завесу, скрывающую ее действия и мотивы, то ей необходимо уделять больше внимания взаимодействию с сообществом. Когда это случится, Linux действительно созреет для предприятий. LXF

› Хотя Стюарт Коэн и сказал, что выражение «центр тяжести» было шуткой, но куда денешься – оно попало даже в последнюю CGL-спецификацию 3.1.

> **«И верно, группа Desktop Linux дает возможность участия для всего Linux-сообщества.»**

Проекты, о которых стоит знать...

OSDL наделала много шума из продвижения Linux на корпоративный рынок, но она не только готовит спецификации. Вот несколько более приземленных проектов:

» Создание Фонда Юридической защиты Linux
В начале 2004 г., когда была серьезная угроза судебных исков со стороны группы SCO ко всем использующим ОС Linux, OSDL создала Фонд Юридической Защиты Linux, выделив $10 млн. для помощи конечным пользователям, втянутым в битву с SCO.

» Сертификат разработчика о Происхождении (Developer's Certificate of Origin)
Другим результатом тяжбы с SCO стало понимание необходимости отслеживания вклада отдельных программистов в разработку ядра. OSDL приняла меры (и не удивительно, коли в ней работают два основных разработчика ядра) и выступила с Сертификатом Разработчика о Происхождении. Он состоит из нескольких параграфов, снабженных цифровой подписью, чтобы корректно описать ваш вклад в древо исходного кода ядра.

» Проект Patent Commons
Законодательство о патентах на ПО изменило лицо разработки приложений в США и вызвало резонанс в остальном мире. Это еще одна угроза будущему разработки приложений с открытым кодом. Проект был создан в августе 2005 г. после внезапного ливня патентов, пожертвованных сообществу Open Source крупными организациями вроде Sun Microsystems, IBM и Nokia. Проект, как и другие инициативы OSDL, разработан для того, чтобы играть роль центра и способствовать использованию открытых патентов на законных основаниях.



Машины — в RAID!



Достала низкая производительность или частые отказы SATA? Объединение ваших жестких дисков может улучшить скорость их работы и повысить надежность, пишет **Нейл Ботвик**.

Кто-то может сказать, что название напоминает оружие массового уничтожения летающих насекомых, но RAID – замечательный способ улучшить производительность вашего компьютера и повысить безопасность ваших данных, используя только ПО, которое у вас уже есть. Если вам нужна лучшая производительность диска, читайте дальше.

Изготовители дисководов снизили сроки гарантии на стандартные ATA- и SATA-диски, и теперь зачастую дают всего лишь год. Выходит, диски стали менее надежными? У нас нет способа узнать это наверняка, но если вы хотите, чтобы ваши данные уцелели в случае проблем с диском, серьезно подумайте о RAID. Этот термин первоначально означал Redundant Array of Inexpensive Disks (избыточный массив недорогих дисков), довольно обманчивое имя, поскольку RAID в те времена был вовсе не дешев. Акроним позже был изменен на Redundant Array of Independent Disks (избыточный массив независимых дисков).

Концепция дискового массива заключается в объединении двух или более устройств (дисков или разделов) так, чтобы они воспринимались системой как одно устройство. Вашими физическими устройствами могут быть **/dev/hda1** и **/dev/hdc1**, но большая часть ОС и приложения увидят **/dev/md0**. С точки зрения системы диск только один. Фактически же это комбинация двух и более устройств с разбросанными по ним вашими данными.

Linux обрабатывает все это, когда ядро только загружается, распознавая, что ваши диски содержат признаки RAID, и создает устройства **/dev/md**. Это происходит на раннем этапе загрузки, до монтирования вашего корневого раздела, так что его тоже можно разместить на RAID. Посмотрим правде в глаза – нет особого смысла в добавлении скорости и безопасности, если это не затрагивает вашу операционную систему.

Пространство, темп, мощь

Использование RAID сперва было нацелено на повышение надежности, поскольку массив сохраняет работоспособность, если один диск выходит из строя, и скорости, поскольку размещение одних и тех же данных на более чем одном диске снижает время доступа и чтения. «Избыточный» не означает «ненужный»; это слово указывает, что данные хранятся более чем на одном диске, и если один из дисков поломается, данные будут по-прежнему в безопасности. Отдельные диски могут считаться избыточными, но не массив в целом. Но если назвать его Array of Redundant Independent Disks (массив избыточных независимых дисков), акроним будет скорее напоминать дезодорант [«arid» по-английски «сухой», – прим. пер.], чем убийцу насекомых. В наши дни RAID также используется и как средство комбинирования различных дисков в один модуль, хотя LVM (см. практикум по этой теме в прошлом выпуске), вероятно, подходит для этого лучше.

Далее мы покажем, как опробовать RAID самому. Вам понадобится пара лишних разделов, желательно на различных дисках. Вы можете попробовать и два раздела на одном диске, но не увидите реальных преимуществ. Но сначала мы слегка углубимся в принципы работы RAID.



Matt Herring

Как работает RAID

Плутон – планета? Курица – птица? RAID 0 – действительно RAID?

Есть несколько уровней RAID, различающихся требованиями к минимальному числу дисков и компромиссом между производительностью, надежностью и дисковой памятью, которые они обеспечивают. Заметьте, что хотя RAID часто используется на серверах с устройствами, допускающими горячее подключение и позволяющими выполнять на лету замещение неисправных приводов, без перерыва в работе, он не сделает ваши диски волшебным образом поддерживающими горячую замену. IDE-диски нельзя заменять «на лету», а SCSI – разве что при наличии соответствующего интерфейса и драйверов. У вас либо есть дисковый контроллер с поддержкой горячей замены, либо нет; RAID не добавит этой способности вашей системе, так что не испытывайте ее. Вы можете остаться со сгоревшим диском/контроллером, или, в технических терминах, с невозможностью входа.

Наряду с реализацией на различных уровнях, RAID может быть аппаратным или программным. Аппаратный RAID-контроллер управляет всем так, что дисковый массив виден как один диск даже на самом низком уровне. Программный RAID, как вы уже догадались, использует ПО вашей операционной системы – у Linux собранное в ядре – для объединения отдельных устройств в RAID-массив. На современных процессорах программный RAID может быть по меньшей мере не медленнее выделенного аппаратного контроллера и, при этом, значительно дешевле.

Есть третий вариант RAID: так называемый аппаратный RAID-контроллер, встраиваемый во многие материнские платы. На самом деле это программная система RAID, маскирующаяся под аппаратный RAID. Ее часто называют гибридным RAID или, менее деликатно, «фальшивым» (fake). Вспомните Win-модемы, и вы поймете идею. Гибридный RAID не так гибок, как программный, он может работать в пределах ограничений BIOS и только на дисках, подключенных к контроллеру, тогда как программный RAID работает с любым диском или разделом, подключенным к любому контроллеру.

```
Session  Edit  View  Bookmarks  Settings  Help
[nelz@hactar RAID]$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid5] [raid4]
md4 : active raid5 sdd6[0] sdd8[2] sdd7[3](F)
      11711104 blocks level 5, 64k chunk, algorithm 2 [3/2] [

md1 : active raid1 sdb7[1] sda7[0]
      369344 blocks [2/2] [UU]

md2 : active raid1 sdb8[1] sda8[0]
      93747200 blocks [2/2] [UU]

md3 : active raid0 sdb9[1] sda9[0]
      298407168 blocks 64k chunks

md0 : active raid1 sdb5[1] sda5[0]
      96256 blocks [2/2] [UU]

unused devices: <none>
[nelz@hactar RAID]$
Shell
```

› Можно использовать отдельные RAID-устройства для каждого раздела, чтобы уравнять надежность, место на диске и скорость.

Развертывание гибкой системы

Поскольку аппаратный RAID управляется контроллером, а гибридного лучше избегать, мы рассмотрим только чисто программный RAID.

Под Linux он объединяет физические разделы в RAID-раздел. Поначалу потребуется кое-какая дополнительная работа, поскольку нужно создать отдельное устройство RAID для каждого раздела, а это, в свою очередь, потребует создания разделов-источников для каждого устройства RAID на физических дисках, зато обеспечит гораздо большую гибкость (см. врезку «Разбиение массивов: возможно ли?» на следующей странице), с различными типами RAID для разных целей. На своей настольной системе вы можете использовать RAID 1 для разделов, где безопасность данных более важна, чем собственно скорость, например, **/usr** и **/home**. Но, допустим, вы также используете множество больших временных файлов, типа ISO-образов DVD-дисков и видеофайлов. Они пойдут на раздел RAID 0, который быстрее и дает в два раза больше места для хранения данных, чем RAID 1. Поскольку это в основном временные файлы, в избыточности, предоставляемой RAID 1, нужды нет.

Программный RAID также позволяет вам использовать диски с различными контроллерами, при желании объединяя IDE, SATA и SCSI. »

Внимание: суперпользователь

Многие команды RAID, приведенные в этой статье, требуют для запуска привилегий суперпользователя (root). Вам нужно открыть терминал и ввести **su -** (**sudo bash** в Ubuntu и других дистрибутивах, основанных на Debian), а затем ваш пароль суперпользователя. Большинство команд RAID бывают разрушительны, так что хорошей идеей является экспериментирование там, где нельзя навредить, например, на отдельном диске.

Также можно создать RAID-массив, используя устройство loop. Его мало используют в реальном мире, но это хороший способ проводить эксперименты, не трогая ваши разделы на жестком диске. Все, что вам понадобится, это достаточное количество свободного места, чтобы создать loop-файлы:

```
dd if=/dev/zero of=raidtest1 bs=1M count=8192
losetup /dev/loop1 raidtest1
dd if=/dev/zero of=raidtest2 bs=1M count=8192
losetup /dev/loop2 raidtest2
```

Эти команды создадут два файла по 8 ГБ и присоединят их к устройствам loop, которые вы сможете использовать в RAID-массиве.

Раскапывая уровни RAID

Уровни 2 и 3 используются редко. RAID-массивы могут быть вложенными, типа RAID 5+0. Если от этой мысли у вас ломит виски, не беспокойтесь – здесь мы такого касаться не будем!

» **RAID 0**. Популярен среди пользователей настольных систем, особенно с «фальшивыми» RAID, но фактически это не RAID, поскольку не обеспечивается избыточность (может, назвать его AID 0?). Записывает данные поочередно на один диск, затем на другой (другие), каковой процесс известен под названием «striping» (страйпинг, чересполосица), он увеличивает производительность. Доступное место складывается из размеров используемых устройств. Сбой одного из дисков испортит весь массив.

» **RAID 1**. Простейший из настоящих систем RAID. Использует два или более устройств, в идеале — одинакового размера. Каждый диск является «зеркалом» другого; если размеры отличаются, массив будет иметь размер наименьшего диска. Дает большую скорость чтения, чем одиночный диск, так как данные можно читать более чем с одного диска одновременно. Основное преимущество – безопасность: если один диск выйдет из строя, RAID продолжит работать, как прежде, используя исправный диск. Каждый диск, добавляемый в массив RAID 1, увеличивает скорость чтения и надежность, но не дисковое пространство.

» **RAID 4**. Немного похож на RAID 0, поскольку размещает свои данные на двух или более дисках, но сверх того использует диск четности. Это отдельный диск, который хранит информацию о записанных данных, чтобы использовать ее для восстановления данных в случае сбоя диска. Это означает, что для RAID 4 требуется как минимум три диска.

» **RAID 5**. Не использует отдельный диск четности, а хранит эту информацию на дисках в RAID. Производительность записи может быть низкой, поскольку данные и информация о четности должны записываться на несколько дисков, но производительность операций чтения высокая. Зачастую это лучший вариант для небольших или средних серверов.

» **RAID 6**. Эволюция RAID 5: предназначен для систем с большими массивами. Предоставляет лучшую защиту от сбоев нескольких дисков в таких массивах, но неэффективен, если используется малое число устройств.



Запускаем управление дисками

Использование инструментов и опций ядра для создания RAID-массива из двух разделов.

Теперь, когда вы знаете, как это работает, давайте подготовим вашу систему для RAID. Прежде всего вам понадобится ядро с поддержкой RAID. Она доступна начиная с версии 2.4, так что единственный вопрос – активирована ли она в вашем ядре (ответ, скорее всего, «да», если вы ставили ядро по умолчанию в одном из стандартных дистрибутивов). Если вы собираете собственное ядро, вам понадобятся следующие опции: Device Drivers, Multi-device support (RAID and LVM), Multiple devices driver support (RAID and LVM) и RAID support. Также возьмите модули того типа RAID, который вы хотите использовать: сойдут raid0, raid1 и raid5.

Вы можете проверить поддержку RAID следующим образом:

```
modprobe raid1 (или тот, модули для которого вы создали)
cat /proc/mdstat
```

Хотя /porc/mdstat должен отчитаться, что у вас нет устройств RAID, само его существование покажет, что RAID готов к использованию.

Для работы с массивами RAID есть два набора инструментов – оригинальный raidtools и более свежий mdadm. С его единственной командой, уместными аргументами и центральной man-страницей, mdadm проще в работе, и здесь мы будем использовать именно его, но перечислим также и команды raidtools во врезке «Справка по командам», справа. Большинство дистрибутивов включают mdadm (обычно и raidtools) на свои установочные диски; некоторые даже устанавливают их по умолчанию.

```
Session  Edit  View  Bookmarks  Settings  Help
[nelz@hactar RAID]$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid5] [raid4]
md4 : active raid5 sdd6[0] sdd8[2] sdd7[3](F)
      11711104 blocks level 5, 64k chunk, algorithm 2 [3/2] [U_U]

md1 : active raid1 sdb7[1] sda7[0]
      369344 blocks [2/2] [UU]

md2 : active raid1 sdb8[1] sda8[0]
      93747200 blocks [2/2] [UU]

md3 : active raid0 sdb9[1] sda9[0]
      298407168 blocks 64k chunks

md0 : active raid1 sdb5[1] sda5[0]
      96256 blocks [2/2] [UU]

unused devices: <none>
[nelz@hactar RAID]$
Shell
```

› Содержимое /proc/mdstat показывает, что происходит с вашими массивами RAID. Здесь у нас смесь RAID 0, 1 и 5 с неисправным диском в массиве RAID 5.

Разбиение массивов: возможно ли?

Программный RAID работает на уровне разделов. Если вы хотите создать массив RAID с четырьмя разделами, вы должны будете сформировать четыре раздела на каждом из ваших дисков и затем создать четыре RAID-массива для каждого из них. Это основной недостаток программных RAID, по крайней мере, с точки зрения администратора. Решение – создать один большой массив и использовать его как физический том LVM, затем создать все ваши разделы как логические тома.

Поскольку за настройки RAID отвечает ядро, а ядро размещается на диске в **/boot**, вам потребуется обеспечить загрузку ядра до запуска RAID. Можно это делать с помощью отдельного раздела **/boot**, не являющегося частью RAID. Есть и другой способ, когда ваша система располагается в RAID на 100%. Начало каждого раздела RAID выглядит как обычный раздел жесткого диска. Так что создайте массив RAID 1 для **/boot**, используя раздел на каждом диске, и ваша система будет просто загружаться с первого диска, настроенного в BIOS.

Что же касается вопроса, где монтировать раздел подкачки (**swap**), то можете сделать **swap** частью вашей LVM-системы, или использовать обычный раздел на каждом диске. Особого смысла в размещении **swap** на RAID нет – ядро Linux в любом случае прекрасно умеет использовать несколько разделов swap, распределяя загрузку между ними. Если один из ваших дисков больше, чем остальные, используйте оставшееся место для размещения **swap**.

Путь RAID 1

Начнем с простейшего примера: создать массива RAID 1, используя два раздела, **/dev/hda1** и **/dev/hdc1**. Запустите ваш любимый инструмент для разбиения диска (попробуйте cfdisk) и пометьте разделы как «Linux raid autodetect (filesystem type FD)». Ядро распознает разделы как компоненты RAID и настроит RAID при загрузке. Это означает, что вы можете поместить свой корневой раздел на RAID.

Настроив разделы, создайте RAID с помощью

```
mdadm --create /dev/md0 --auto=yes --level=raid1 --raid-devices=2 /dev/hda1 /dev/hdc1
```

Эта команда создаст массив в **/dev/md0** («md» – сокращение от Multi-Device): опция **auto** дает указание создать также и узел (node) устройства. Уровень – RAID 1, с двумя устройствами: **hda1** и **hdc1**. Если вы хотите использовать более двух устройств, допишите дополнительные в командную строку и измените **--raid-devices** соответствующим образом. Число устройств должно равняться значению **--raid-devices** (плюс **--spare-devices**, но об этом чуть позже).

Если какой-то из разделов содержит файловую систему, *mdadm*, прежде чем продолжить работу, предупредит вас, что вы можете частично потерять ваши данные. Файловая система на первом разделе, заданном в командной строке, будет сохранена на RAID, но остальные будут уничтожены (при других уровнях RAID пропадут все разделы).

Наберите **cat /proc/mdstat**, чтобы убедиться, что ваш массив создан. Вы увидите нечто вроде

```
md0: active raid1 hdc1[1] hda1[0]
      11719296 blocks [2/2] [UU]
      [=>.................] resync = 5.6% (665024/11719296)
finish=43.5 min speed=4227K/sec
```

Массив синхронизирован. Данные с первого диска скопированы на второй. Этот процесс будет протекать в фоновом режиме, и если первый диск содержал файловую систему, вы сможете смонтировать его и начать работу с массивом немедленно:

```
mount /dev/md0 /mnt/somewhere
ls -l /mnt/somewhere
```

Если разделы новые, вам сначала потребуется создать файловую систему:

```
mke2fs -j /dev/md0
mount /dev/md0 /mnt/newstuff
cp -a /mnt/oldstuff/* /mnt/newstuff
```

Команда для создания массива RAID 0 та же самая, нужно лишь поменять уровень на raid0. Любая существующая файловая система будет разрушена, так что вам нужно будет создать файловую систему на устройстве, прежде чем вы сможете с ним что-нибудь сделать. Чтобы создать массив RAID 5, потребуется минимум три диска, в остальном команда та же самая, с очевидным отличием в параметре **--level**.

Управление системой

Что делать, если диск вышел из строя, а вы за 100 миль от него; а также другие заботы администратора.

Итак, вы настроили RAID, установили свои файлы, и все работает прекрасно. Возможно, вы отслеживаете устройства массивов и компонентов с помощью полезных опций mdadm: --query, --detail и --examine. Одна из основных причин для перехода на RAID 1 – функция избыточности; так что же будет, если одному из дисков «поплохеет»?

Проще всего ответить в случае RAID 0, поскольку здесь избыточности нет. Если один диск вышел из строя, это означает несостоятельность массива. Ваши файлы погибли, так что доставайте резервные копии – да-да, те самые, которые вы собирались обновить в прошлом месяце!

С помощью RAID 1 и выше, массив сможет работать без неисправного диска, но со сниженной производительностью. Это лучше, чем лишиться всей системы, но вы вряд ли захотите работать в таком режиме сверх необходимого, поскольку вы беззащитны перед поломкой и второго диска вслед за первым. (Не столь невероятной, как кажется: бывают сбои по внешним причинам, типа скачка напряжения).

Предположим, у вас есть массив RAID 1 на **sda1** и **sdb1**, и **sdb1** выходит из строя. Прежде всего нужно удалить неисправный диск из массива. Он уже должен быть помечен как аварийный – для проверки используйте **cat /proc/mdstat**, диск будет отмечен как (F) – так что удалите его командой

```
mdadm /dev/md0 --fail /dev/sdb1 --remove /dev/sdb1
```

Теперь можно отключить питание, снять неисправный диск и заменить его другим, должным образом разбитым на разделы, как мы вам показывали. Когда вы загрузитесь, **/dev/md0** будет продолжать работать в деградированном режиме. Тут-то и добавьте новый диск:

```
mdadm /dev/md0 --add /dev/sdb1
```

Массив немедленно приступит к его пересозданию в фоновом режиме. Вы можете ощутить некоторое снижение производительности, но пересоздание работает с низким приоритетом. Как обычно, **/proc/mdstat** будет сообщать вам, что происходит. Отключения питания, добавления диска и ручного ввода команды можно избежать – в составе *mdadm* есть демон, который наблюдает за вашими дисками в фоновом режиме и отправит вам сообщение по электронной почте в случае проблем (не будете же вы ежеминутно проверять **/proc/mdstat**).

Справка по командам

Задача	Команда *mdadm*	Команда *raidtools*
Создать массив	mdadm --create	mkraid [1]
Считать информацию о массиве	mdadm --detail	lsraid
Изменить существующий массив	mdadm --grow	raidreconf [1]
Запустить массив	mdadm --assemble	raidstart
Остановить массив	mdadm --stop	raidstop

[1] Эти программы raidtools требуют, чтобы вы сначала ввели информацию в файл конфигурации. Файл конфигурации raidtools – **/etc/raidtab**, он является обязательным. Файл конфигурации *mdadm* – **/etc/mdadm.conf**; перечисление в нем устройств не обязательно, опция **--scan** для *mdadm* прочитает всю информацию об устройствах сама.

Демон по вызову

В режиме демона, если вы добавили запасной (spare) диск к массиву, этот диск подключится автоматически, когда один из активных дисков выйдет из строя. Вы можете добавить его при создании массива, например:

```
mdadm --create /dev/md0 --auto=yes --level=raid5 --raid-devices=3 --
spare-devices=1 /dev/sda1 /dev/sdb1 /dev/sdc1 /dev/sdd1
```

Эта команда создаст массив RAID 5 с тремя активными дисками и одним запасным. Запасные диски можно использовать на всех уровнях, кроме RAID 0. В уже существующий массив запасное устройство добавляется с помощью

```
mdadm /dev/md0 --add /dev/sdd1
```

Если в массиве есть запасной диск, а в одном из активных дисков обнаружена неисправность, неисправный диск автоматически будет удален из массива, а запасной – добавлен.

Если у вас множество массивов, предоставление запасного диска для каждого из них может быть излишним. Взгляните на опцию **spare-group** на man-странице **mdadm.conf**. Если она установлена, то *mdadm*, определив неисправность диска, передаст неиспользуемый в каком-либо массиве запасной диск нуждающемуся массиву. Эта программа обучаема.

Демон *mdadm* запускается обычным способом, с помощью сценария в **/etc/init.d**. Используйте инструмент управления сервисами вашего дистрибутива для добавления его на стандартный уровень исполнения (runlevel), чтобы запуск происходил всегда. Адрес электронной почты, куда будут отсылаться оповещения, устанавливается добавкой строки в **/etc/mdadm.conf**:

```
MAILADDR me@mydomain.com
```

Можете также заставить демона запускать какую-либо программу в ответ на «интересное событие» (эвфемизм, применяемый man-страницей для сбоя): например, отсылать вам IM- или текстовое сообщение через SMS-шлюз. Выбор за вами, хотя бешеные вспышки и динамики, орущие «Атас, Вован!», многие сочтут перебором.

Ничто не защитит вас от сбоя нескольких дисков, так что, получив письмо о сбое диска, поскорее замените его, пока по закону Мэрфи не поломался другой.

Можно еще многое рассказать о программных RAID на Linux (man-страница mdadm – хороший источник), но для начала работы мы дали вам более чем достаточно. Только знайте, что вы забавляетесь с данными на вашем жестком диске на низшем уровне, так что будьте осторожны и помните, что RAID не заменяет резервных копий! LXF

Переходим в GUI

Хотя RAID'ом обычно управляют из командной строки, лучшим графическим инструментом его администрирования является вездесущий *Webmin*. Он доступен в репозиториях большинства дистрибутивов, и многие уже установили его на своем компьютере. Вы найдете модуль RAID в секции Hardware. Он запускается как web-сервис, доступный через любой браузер.



› Webmin: просматривайте и управляйте массивами RAID на удаленном сервере или локальном компьютере – даже без графического интерфейса.

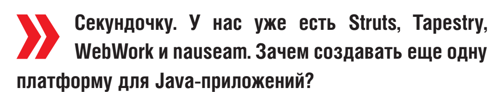
Что за штука... JavaServer Faces?

Ричард Драммонд прослышал о более простом и быстром способе строительства web-приложений. Определенно стоит разобраться…



Сейчас много шумят об этой новой web-технологии, JavaServer Faces. Для чего она?

JavaServer Faces (сокращенно JSF) – компонентная платформа, призванная упростить разработку пользовательских интерфейсов для web-приложений Java. Это…



Секундочку. У нас уже есть Struts, Tapestry, WebWork и nauseam. Зачем создавать еще одну платформу для Java-приложений?

В том-то и часть проблемы: много несовместимых платформ. Однако JSF – открытый стандарт, разрабатываемый в рамках Java Community Process, и представители многих популярных, но нестандартных платформ внесли свою лепту в разработку спецификации JSF (текущая версия JSF 1.2). Спецификация открытая, и применять ее может каждый. Sun, кстати, выпустила стандартную реализацию.

Ясно. Но чем JSF отличается от, скажем, Jakarta Struts?

Тем, что у этих проектов разные цели. Struts – завершенная платформа, позволяющая упростить построение Java-приложений с использованием парадигмы модель-вид-контроллер (MVC). JSF сосредоточена на одной части парадигмы, виде.

Вид — то есть вид web-приложений, отсюда и «Faces»?

Точно. MVC – программная архитектура, четко разделяющая пользовательское приложение на три части: Модель (часть программы, содержащая данные), Вид (представление данных, позволяющее пользователю взаимодействовать с программой) и Управление (обработка событий, например, пользовательского ввода, и обновление в соответствии с этими событиями Модели и/или Вида). Архитектура MVC успешно используется некоторыми современными средами разработки, включая *Cocoa* для Mac OS X, Microsoft Foundation Classes и *Swing* для Java. Теперь наблюдается, что подобный подход начал применяться и к web-приложениям, отсюда и популярность Struts.

Спасибо за объяснение технологии, но что MVC даст лично мне?

Если рассматривать приложение таким образом, можно отделить модель от вида. Это особенно актуально для web-приложений. Вспомните только, как часто популярные сайты подвергаются визуальному обновлению (меняется вид), а обрабатываемые ими данные (модель) остаются неизменными. Кроме того, в web-разработках и у программиста, и у дизайнера (который не обязательно разбирается в программировании) должна быть возможность трудиться над приложением вместе. Наконец, вам могут понадобиться различные методы доставки web-приложения пользователям: скажем, вы работаете с HTML-интерфейсом для настольных компьютеров, а с WML – для мобильных устройств. Благодаря MVC решение этих задач существенно упрощается.

Если JSF занимается только видом, чем она лучше Struts, которая умеет все?

Не обязательно быть во всем лучше всех. JSF лучше только для построения пользовательских интерфейсов. Вполне возможно совместное использование Struts и JSF в одном приложении – есть даже дочерний проект Struts под названием Shale, для более плотной интеграции с JSF (см. http://shale.apache.org).

Об этом хватит! Расскажите подробнее о JSF.

Вижу, вас надо хорошенько убедить. Ладно. Иногда JSF называют «*Swing* для web-приложений», и это правильно (как известно, *Swing* – стандартная среда разработки графических пользовательских интерфейсов, GUI, на чистом Java). Со *Swing* легко изменять способы отображения компонентов интерфейса или создавать особые компоненты, а также использовать визуальные редакторы интерфейсов. Все эти преимущества применимы к JSF, но не к Struts.

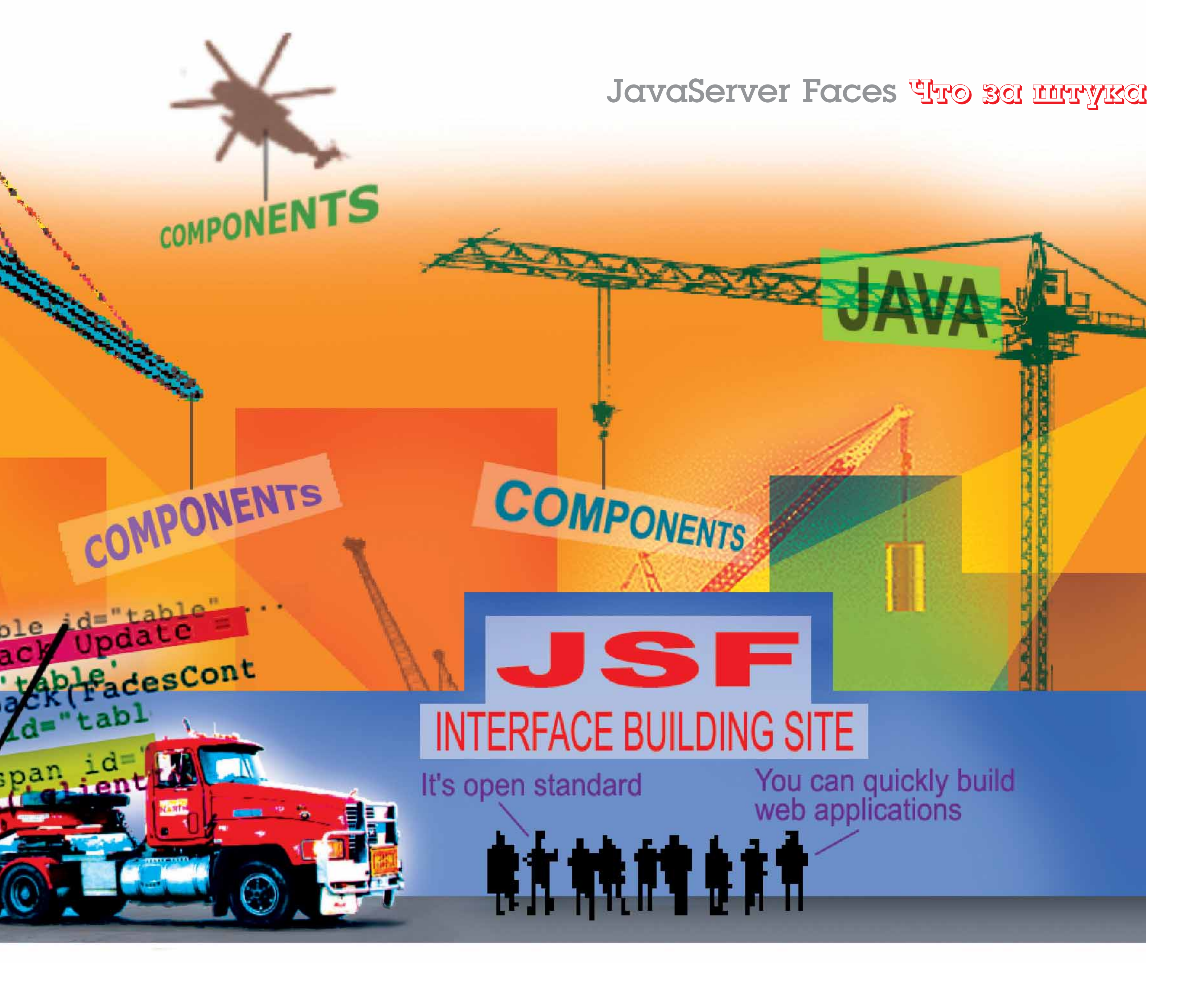


Вы приводите сравнение со Swing. Это, по-вашему, убедительно?

Да забудьте вы эту *Swing*-фобию! Ведь вы же сможете строить интерфейс web-приложения так же легко, как строится простой GUI – перетаскиванием компонентов в UI-конструктор; это ли не здорово?

Допустим. Но интерфейс для web и для обычных платформ – не одно и то же. Осуществима ли эта идея?

Да, поскольку JSF моделирует интерфейсы точно так же, как традиционные среды разработки, в частности, *Swing*. В основе JSF лежит API, позволяющий создавать компоненты интерфейса, не зависящие от устройства вывода. Этот API управляет всеми вопросами, связанными с пользовательским интерфейсом: жизненным циклом, обработкой событий, персистентностью (способность сохранять и поддерживать объекты) и навигацией. JSF предоставляет также базовый набор компонентов GUI, знакомых дизайнеру любых интерфейсов.



Как это отразится на HTML-интерфейсе, который я вижу в браузере?

Основные компоненты интерфейса JSF существуют на серверной стороне приложения и не зависят от средств представления. Однако нынешняя спецификация JSF требует применения рендерера, способного отображать компоненты пользовательского интерфейса как HTML, через JavaServer Pages (JSP). Возможны и другие технологии представления, но поддержка JSP обязательна.

Погодите продолжать, поясните, пожалуйста, что такое JavaServer Pages?

С удовольствием. Основа технологии генерации динамического web-контента с сервера Java-приложений – это сервлет. Но строительство пользовательского интерфейса только через сервлеты весьма проблематично: встраивать весь генерируемый контент (в основном HTML) внутрь Java-кода – задача непосильная. Поэтому и появилась JavaServer Pages. Теперь все наоборот: можно добавить в HTML особые тэги, исполняющие необходимый для генерации динамического контента Java-код, а то и встроить этот код непосредственно в HTML-страницу.

А применение JSF обязательно требует поддержки JSP?

Да, поставляется библиотека тэгов JSP с особыми тэгами, позволяющими оперировать с компонентами пользовательского интерфейса JSF. Например, если вы хотите представить пользователю данные в виде таблицы, нужно воспользоваться соответствующим JSF-тэгом. JSF возьмет на себя генерацию таблицы в виде HTML-кода в ответ на пользовательский запрос, а web-контейнер доставит ее в браузер пользователя.

Полагаю, ввод данных в страницы JSF не составит труда...

К сожалению, избежать Java-кодирования не получится. Каждая страница JSF связана с «бином» (bean), который передает значения компонентам JSF, занимается обработкой событий, проверкой введенных данных и т.п. Каждый компонент JSF связывается со свойствами и методами «бина» через атрибуты тэгов JSF

ОК, вы заинтриговали меня. С чего же начать разработку в JSF?

Можно начать разработку с помощью web-контейнера типа *Tomcat*, вручную инсталлировать экземпляр JSF, затем в простом текстовом редакторе создать шаблоны и необходимый Java-код. Однако простейший путь – зарегистрироваться в Sun Developer Network и установить новейшую версию *Sun Studio Creator* (см. http:// developers.sun.com/prodtech/javatools/jscreator). Это полный пакет для JSF-разработки, включающий Java IDE, визуальный конструктор и сервер приложений. Регистрация свободная, а загрузка *Creator* для зарегистрированных членов бесплатна.

Хм. А нет ли других решений среди свободного ПО?

Creator от Sun основывается на открытом *NetBeans* IDE, но в текущей стабильной версии *NetBeans* еще нет визуальных инструментов для работы с JSF.

Вы сказали, что JSF – открытый стандарт. Нет ли других вариантов, кроме как от Sun?

Конечно, есть: например, *Apache MyFaces* (см. http://myfaces. apache.org).

Хотелось бы познакомиться с библиотеками компонентов JSF.

Отлично! Попробуйте *ADF Faces от Oracle* (www.oracle.com/technology/products/jdev/htdocs/partners/addins/exchange/jsf).

А где можно подробнее узнать о JavaServer Pages?

Рекомендую http://java.sun.com/javaee/javaserverfaces. Там можно найти информацию и ссылки на многочисленные статьи и руководства по JSF. LXF

Линуксцентр представляет авторизованные курсы Mandriva Linux.

Курсы не привязаны к конкретному дистрибутиву и сертифицированы международной организацией LPI - Linux Professional Institute (www.lpi.org).



LNX70: Введение в Linux.

Предназначен для пользователей, которые хотят приобрести навыки использования Linux. В рамках курса рассматриваются вопросы миграции на Linux и даются базовые знания для пользователей.

Цель курса: Обучить пользователя оптимальной работе с Linux.
Требования к соискателям: Базовая компьютерная подготовка.



LPI101: Администрирование Linux для начинающих.

Курс предназначен для пользователей, которые хотят приобрести навыки администрирования Linux. Рассматриваются следующие вопросы: конфигурирование системы, настройка рабочего окружения, основные команды GNU и Unix, файловые системы и графическая среда X-Window.

Цель курса: Подготовить системного администратора Linux в соответствии с требованиями к соискателям на сдачу экзамена LPI101.

Требования к соискателям: Знания по Linux в объеме курса LNX70.



LPI201: Администрирование Linux для специалистов.

Курс предназначен для системных администраторов, которые хотят повысить свою квалификацию в области управления сложными Linux-системами. Рассматриваются следующие вопросы: настройка ядра Linux, восстановление системы, восстановление файловой системы, настройка RAID, сопровождение системы, автоматизация задач администрирования.

Цель курса: Подготовить системного администратора к управлению сложными Linux-системами в соответствии с требованиями к соискателям на сдачу экзамена LPI201.

Требования к соискателям: Знания по Linux в объеме курса LPI101 или LPI102.



LPI102: Сетевое администрирование Linux.

Курс предназначен для пользователей, которые хотят приобрести навыки администрирования сетей на базе Linux. Рассматриваются следующие вопросы: конфигурирование периферийных устройств, настройка рабочего окружения, основные команды GNU и Unix, файловые системы и графическая среда X-Window.

Цель курса: Подготовить системного администратора к управлению сетями на базе Linux в соответствии с требованиями к соискателям на сдачу экзамена LPI102.

Требования к соискателям: Знания по Linux в объеме курса LPI101.



LPI202: Сетевое администрирование Linux для специалистов.

Курс предназначен для опытных системных администраторов, которые хотят усовершенствовать навыки управлениями сетевыми сервисами на базе Linux. Рассматриваются следующие вопросы: углубленное конфигурирование сети, настройка служб электронной почты и новостей, расширенная настройка DNS, обслуживание интернет-сервера, настройка DHCP, NIS, LDAP и PAM, управление безопасностью сетевых служб, контроль за трафиком.

Цель курса: Подготовить системного администратора к управлению сложными сетевыми сервисами в соответствии с требованиями к соискателям на сдачу экзамена LPI202.

Требования к соискателям: Знания по Linux в объеме курса LPI101.

Узнайте подробности и закажите курсы на сайтах - www.linuxcenter.ru и www.unixedu.ru



LINUX FORMAT Учебники

НАШИ ЭКСПЕРТЫ ПОМОГУТ ВАМ С ЛЮБЫМ ПРИЛОЖЕНИЕМ LINUX



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Free! Music

Не кодом единым жив человек. Свободные исходники нужны не только для программ. Музыка тоже достойна быть свободной. Писать для людей бывает интереснее, чем ради денег.

Есть люди, из которых музыка исходит. Это своеобразный талант – как дар математика. Их мало, как и математиков, но они есть. Я с подобными личностями пересекался и подобные встречи укрепляют мою веру в человечество. Вовсе необязательно создавать шедевры всех времен и народов, ведь понятие шедевра относительно.

> «Музыку раньше писали для людей, а теперь для шоу-бизнеса.»
> **Валерий Мифодовский**

Сайт http://freemusic.org.ru/ создан для таких людей. Любой может выставить там свое произведение для свободного прослушивания и тиражирования под свободной лицензией. У сайта нет большого и активного сообщества, но вполне возможно, что за подобными начинаниями будущее.

У свободной музыки, как и у движения Open Source, очень схожие проблемы. И там, и там могучие структуры трясутся за свои гроши и насильно навязывают целым народам только этим структурам выгодные решения. Но уже поздно — свободное творчество полностью не задавишь.

Я не ценитель музыки вследствие отсутствия музыкального слуха (проверено шестью годами в музыкальной школе), но, по-моему, свободная музыка просто обречена занять свое место в умах и сердцах разумных людей. Рынок, естественно, завоеван не будет, но для желающих всегда найдется отдушина.

E.M.Baldin@inp.nsk.su

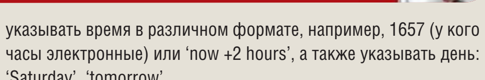
В этом выпуске...

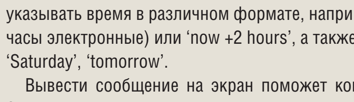


СОВЕТ МЕСЯЦА: Будильник

Как известно, компьютеры нужны для выполнения рутинных операций: сложения и вычитания, рисования разными цветами и уведомления своих всесильных хозяев о надвигающейся встрече. Существуют десятки приложений, которые выскакивают в самый неподходящий момент и утверждают, что вы забыли посетить стоматолога. На самом деле, все они не нужны – пара команд, запущенных со старого доброго терминала, выполнит всю работу. Начнем, пожалуй с *at*.

Команда *at* – одна из тех, к которым рано или поздно приходят все пользователи Linux. Наберите at **4:57 today** – откроется примитивный текстовый редактор. Введите в нем все команды, которые желаете выполнить в 4:57, а затем нажмите Ctrl-D, чтобы отметить конец файла. Вы можете указывать время в различном формате, например, 1657 (у кого часы электронные) или 'now +2 hours', а также указывать день: 'Saturday', 'tomorrow'.





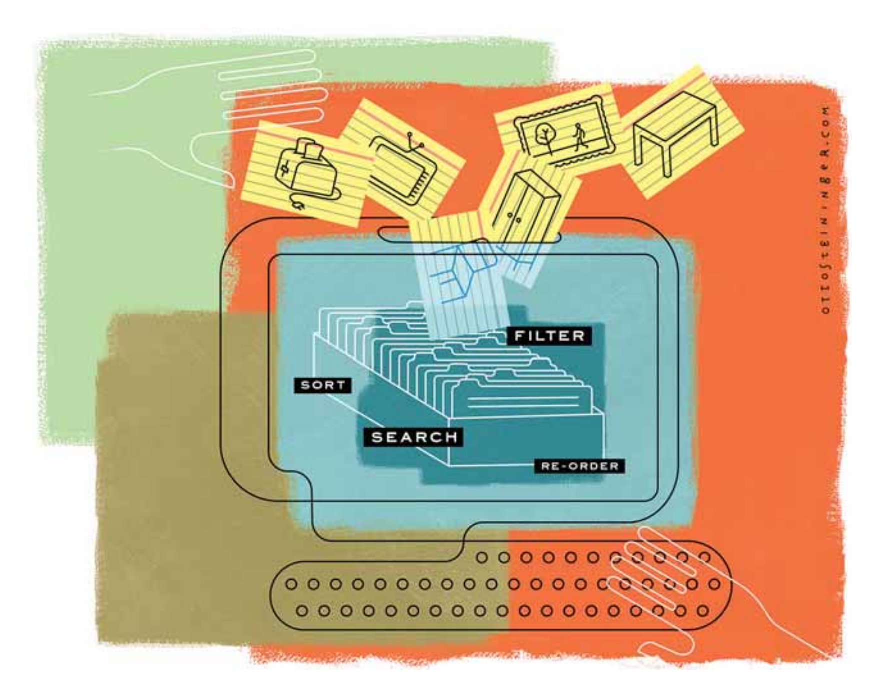
Вывести сообщение на экран поможет команда *xmessage*. Это простая утилита, которая уходит корнями в темное прошлое X Window; она открывает диалоговое окно с сообщением и кнопкой «Закрыть». Синтаксис *xmessage* тоже не слишком сложен:

```
xmessage -display :0.0 "Ты опоздал!"
```

Важно не забыть параметр **display**, потому что иначе *xmessage* ни за что не догадается, какой экран ей использовать. Скрестив *xmessage* с *at*, вы получите мощный инструмент, не прибегая к сторонним приложениям.

OOo Base: Шаг 1

Базы данных – это сложные непонятные штуки, в которых разбираются только крутые хакеры? Это так, да не совсем! **Энди Ченнел** берет быка за рога и принимается за *OpenOffice.org Base*.



Для многих обычных пользователей базы данных не представляют интереса: они выглядят чересчур сложными для того, чтобы стоило возиться с ними. Однако существует пара приложений, которые пытаются устранить этот барьер и принести радость (я действительно сказал «радость»?) работы с БД в массы. Последняя версия *KOffice* включает в себя неплохое приложение *Kexi*, в то время как офисный пакет *OpenOffice.org* укомплектован программой под названием *Base*. В этот раз мы поговорим именно о последней, потому что, с одной стороны, она не так сильно интегрирована в рабочее окружение, как *Kexi* (что конечно не мешает вам использовать *Kexi* в Gnome), а с другой, *OOo* в целом справляется с поставленными задачами чуть лучше, что будет для нас важно на начальных стадиях нашего проекта.

На самом деле, база данных, которую я собираюсь создать представляет собой инвентаризацию моих вещей на тот случай, если они будут украдены или просто потеряются. Для этого в *OOo* Base имеется специальный мастер – по сути, это заранее написанный скрипт – как раз для моей цели. Итак, мы начнем с этого примера, который иллюстрирует важные для нас принципы, а затем немного изменим стандартную БД, углубившись в теорию.

На нашем диске вы найдете последнюю версию OOo, которая содержит готовый к работе Base и остальные компоненты этого офисного пакета.

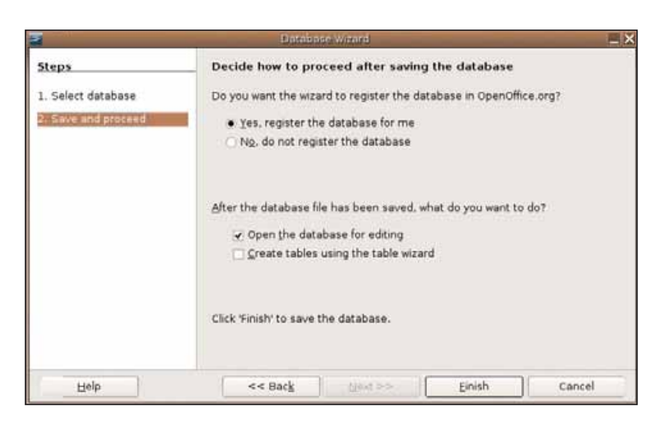


Наш эксперт

Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется еще со времен Dragon 32.

Часть 1 Что такое база данных?

Давайте сперва запустим приложение, для чего выберите *OpenOffice.org Base* из пункта меню Офис (в вашем дистрибутиве он может называться по-другому). Когда появится окно программы, выберите пункт Create a New Database (Создать новую базу данных). На следующем экране убедитесь что оба пункта Yes, register the database for me (Да, зарегистрировать мне базу данных) и Open the database for editing (Открыть базу для редактирования) отмечены, после чего нажимайте кнопку Finish.



› Регистрация БД уведомит *OOo* об ее существовании. Это полезно, если вы, к примеру, создаете документы для большой адресной книги.

Работа по созданию БД обычно состоит из двух частей: структурирование и наполнение. Последнее целиком зависит от вас, поэтому мы сосредоточимся на структуре, которая изначально задается в разделе Таблицы.

На что способны таблицы

Каждый, кто работал с электронными таблицами, будет чувствовать себя комфортно при виде столбцов, строк и ячеек в базе данных. На самом деле, обычные электронные таблицы даже предпочтительнее использовать в том случае, если вы работаете с небольшим объемом данных: они легки в понимании, настройке и последующем визуальном редактировании всей структуры записей.

Преимущества базы данных проявляются в тот момент, когда нам нужно объединять информацию из разных источников и работать – при необходимости – с большими массивами данных, которые при этом не будут загружаться в память компьютера все сразу. К тому же, гораздо проще создать массив данных при помощи специальной «формы» с текстовыми полями, флажками и переключателями, чем исполь-

» **Месяц назад** Мы создали по-настоящему детский рабочий стол с ограничением доступа и web-фильтрами



зуя обычные таблицы с колонками. В БД мы также можем создавать связи и отношения между массивами, что помогает управиться с большим объемом данных.

Убедились? Теперь перейдем к таблицам, и запомните: девиз серии «Первые шаги» – Сначала составляем план.

При создании базы данных перво-наперво нужно понять, из чего она будет состоять и какие связи между объектами вы собираетесь установить (если вообще собираетесь). Для начала лучше всего взять лист бумаги и написать на нем список вещей, которые вы собираетесь включить в БД. В нашем случае речь идет о списке домашних вещей, поэтому такие параметры как название предмета, цена и дата покупки имеют решающее значение. Мы собираемся создать очень простой массив данных, поэтому эти предметы будут «полями» БД, в которые мы и внесем детали.

Определяемся с данными

Нужно продумать, какого типа будет каждая информационная ячейка. Поле Product Name (Название предмета) [кстати, в БД предпочтительнее использовать названия, набранные латинскими буквами. Это не относится к тексту ячеек, – прим.ред.] должно быть текстовым, в то время как поле Price (цена) – числовым, причем в определенном формате – десятичном или валютном – для того, чтобы введенные цены автоматически приводились к одному виду.

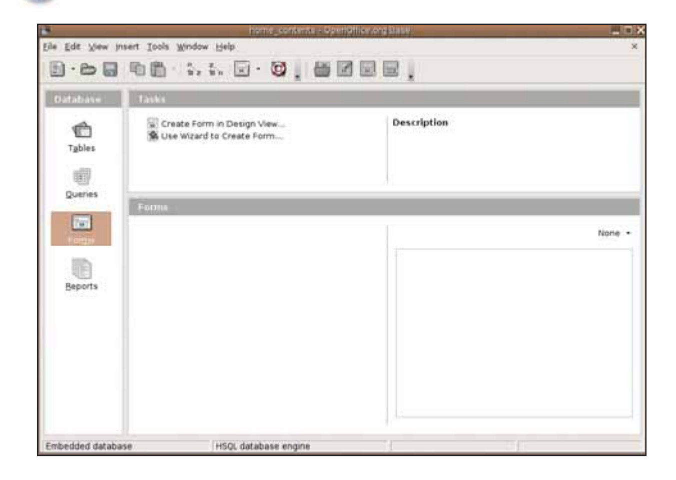
На первое время мы будем использовать всего несколько типов данных, включая простой текст, десятичные числа и дату, но в дальнейшем можно будет добавить и отдельные категории (кухня, компьютер, музыка и др.) во вторую таблицу. В нее, в свою очередь, можно включить параметр «да/нет», указывающий на то, использует ли категория флажки или выпадающие списки. Более крупная БД вполне может содержать две и более таблиц. Например, база данных школы может содержать одну таблицу с информацией об учителях и вторую – о детях; эти две таблицы могут быть связаны отношением учителя к классу, в котором он преподает. Третья таблица может быть посвящена оценкам за экзамены и при этом быть связанной с таблицами учителей и учеников.

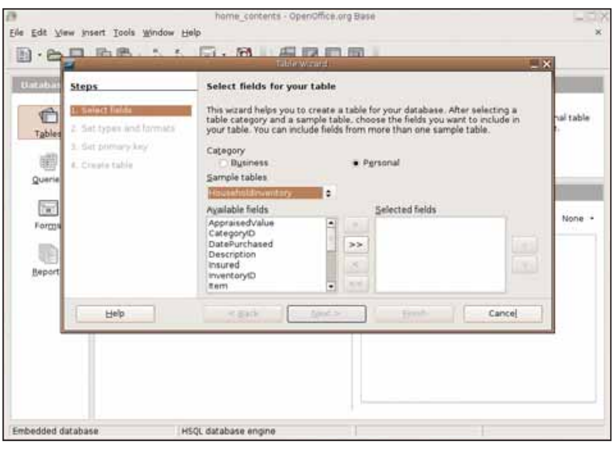
Сейчас у меня есть список из семи свойств предметов: название, производитель, модель, цена, место покупки, дата покупки и серийный номер. Эти свойства следует объединить под общей категорией Record ID (Карточка), поэтому самое время запустить первый мастер в *OOo Base*: мастер таблиц.

Вниманию грабителей

Хранить всю важную информацию на компьютере – достаточно легкомысленная затея, особенно если учесть, что компьютер в первую очередь привлекает внимание квартирных воров. В то же время, нельзя забывать и про резервные копии. Здесь вам не помешает немного параноидальной осторожности и совет хранить резервные CD/DVD в каком-нибудь месте вне дома. К счастью, наша простая база данных представляет собой один аккуратный ODB-файл, который можно отправить по e-mail на удаленный web-сервер или записать на CD-R и спрятать на дедушкином чердаке.

Часть 2 Немного волшебства





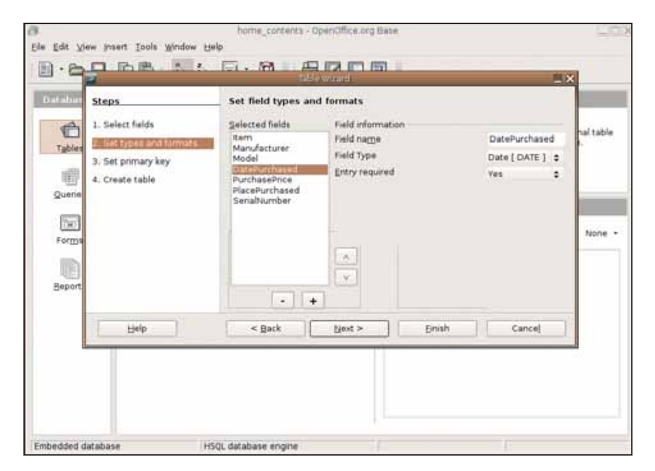
1 Вызовите мастера

В главном окне вашей БД щелкните по кнопке Таблицы слева и выберите пункт «Использовать мастер для создания таблицы», после чего откроется окно мастера. Если, к примеру, вы хотите упорядочить не домашнюю утварь, а что-то другое, для чего у OOo нет мастера, выберите пункт Создать таблицу в режиме дизайна. Создание таблицы одинаково и там, и там, просто в последнем случае придется кое-что дописать вручную.

2 Выберите поля

Начните с выбора наиболее подходящей для вас категории – Деловой или Персональной, в соответствии с которыми меняется содержимое примеров таблиц и списка полей. Нас интересует категория Персональный и тип таблицы под названием Домашний_Инвентарь. Выбрав эти пункты, мы переходим чуть ниже, и теперь нам нужно переместить нужные нам переменные поля из левой части списка в правую. Для этого щелкните по названию поля, а затем по верхней кнопке со стрелкой вправо (кнопка с двойной стрелкой работает по принципу «выделить все»). Выбранные вами поля можно сдвигать относительно друг друга при помощи кнопок вверх/вниз, меняя таким образом их очередность.

»



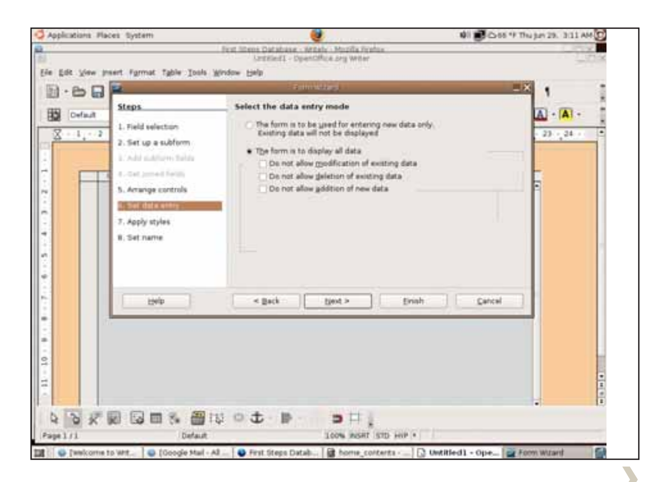
3 Что за данные?

На следующем этапе вам нужно задать тип данных для каждого поля. Почти везде у нас будут текстовые поля, но нам также понадобится одно числовое поле (для цены) и одно поле в формате даты (для времени покупки). Для каждого поля есть специальный пункт, определяющий обязательность его заполнения. Если выставить значение ***да***, то пользователь не сможет сохранить изменения до тех пор, пока не заполнит это поле. На этом этапе вы также может добавить, удалить или изменить порядок полей, используя кнопки +, – и стрелки вверх/вниз. Если вы не пользуетесь мастером, два последних действия можно выполнить в отдельном окне, которое по виду похоже на электронную таблицу. Слева вы вводите названия полей, а справа выбираете из списка нуж-



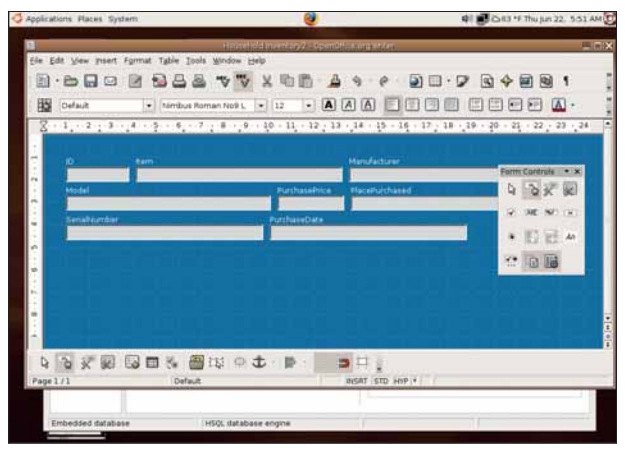
4 Создаем форму

Следующим шагом будет запуск мастера форм. Форма делает именно то, что подразумевает ее имя: отображает специальные области, которые пользователь должен заполнить, после чего вставляет полученные данные в таблицу. Выберите в левой части окна кнопку Формы, а затем пункт Использовать мастер для создания формы. Вы увидите окно, очень похожее на мастер таблиц, в котором также будет список полей, которые на этот раз можно включить в состав формы, используя уже описанные кнопки со стрелками. Просто перенесите в правый список нужные поля один за другим. Следующий шаг мастера мы пропустим, так как у нас нет подформ (это вложенные формы, которые можно подсоединить, к примеру, ко второй таблице).



5 Ограничения полей ввода

Теперь пора выбрать стиль формы – расположение полей ввода и меток. Щелкните по нужному варианту и перейдите на следующий экран, отвечающий за источники данных. На этом этапе мы можем установить ограничения на дальнейшее редактирование содержимого полей. Из двух категорий в этом окне нас интересует вторая (Форма для отображения всех данных), которая означает, что все пользователи смогут просматривать содержимое БД через диалоги форм OOo. Оставьте все пункты нетронутыми.



6 Последний шаг – оформление

Наконец у нас появилась возможность выбрать стиль оформления из подборки *OOo*, дать форме имя и насладиться результатом – готовой для ввода данных формой! Особо отмечу, что для своей БД вы можете наделать сотни таких форм, просто запуская мастер снова и снова. Это может сыграть вам на руку, если вы хотите создать две почти одинаковые формы, в одну из которых будет внесена, к примеру, стоимость компьютеров, гитар и редких пластинок, а в другую – нет. Так вы сможете сохранить конфиденциальность своих расходов. И не только вы.

Бриджит Бардо использует Mandriva

Ну каким еще заголовком привлечь вас к чтению о моделях БД? Так что, пока вы здесь... В Wikipedia описаны шесть популярных типов баз данных, однако действительно широко распространены только два из них: плоский файл и реляционная БД. Первые представляют собой двумерный массив строк и столбцов. Строки связаны между собой – они описывают одну и ту же сущность, а колонки содержат данные определенного типа: валюту, дату и т.п. База данных из нашего учебника – типичный пример плоского файла. В свою очередь, реляционная БД состоит из нескольких плоских файлов (таблиц) и хорошо подходит для описания взаимоотношений между ними. Распределение данных между несколькими таблицами снижает число дублей и позволяет улавливать гораздо более глубокие связи. Например, результаты экзамена можно связать с учителем, группой учащихся и Министерством Образования.

Часть 3 Добавляем данные

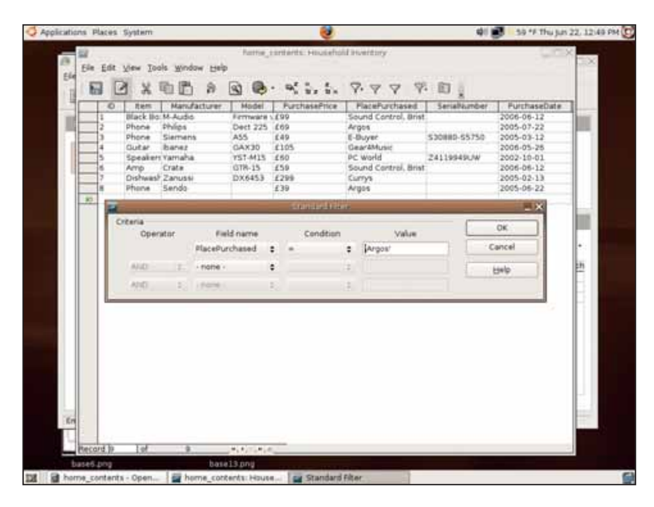
Два мастера помогли нам подготовить основу для внесения информации, и теперь мы как раз можем этим заняться. Для этого есть два способа. Первый заключается в заполнении таблиц(ы): перейдите в соответствующий раздел на панели слева, щелкните правой кнопкой мыши на созданной таблице и выберите Открыть. Вы окажетесь в редакторе таблиц, столбцы которого будут соответствовать названиям полей. Данные можно вносить в пустые ячейки. Щелкните по ячейке, наберите какой-нибудь текст и нажмите Tab, чтобы перескочить на следующую. Продолжайте, пока не кончатся данные. Вы можете заметить, что когда вы набираете содержимое поля, под ним автоматически появляется новая пустая строка, также готовая для ввода данных. Когда вы дойдете до последней ячейки в строке, нажатие Tab перенесет вас в первую ячейку следующей строки.

А как же второй способ заполнения базы данных? Об этом мы поговорим в следующий раз...

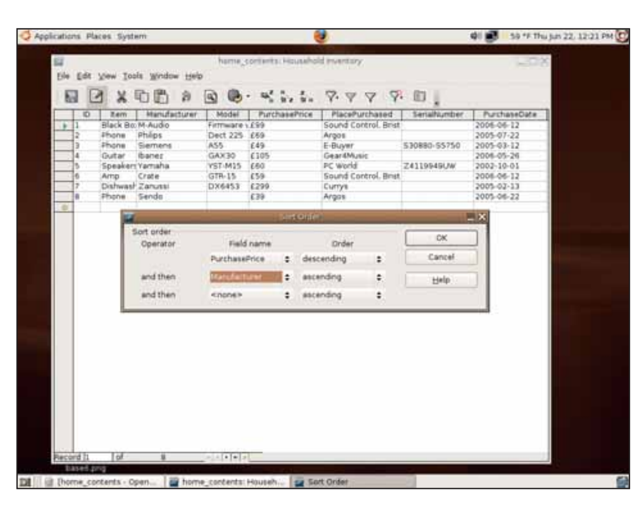
Пойдем со мной

В редакторе таблиц имеются некоторые важные инструменты, позволяющие делать большую часть работы в этом окне, поэтому давайте изучим его подробнее. Вдоль верхней части окна располагается строка меню с привычными пунктами. Под ней находится панель инструментов, с помощью которых можно фильтровать, проводить поиск и менять порядок полей в вашей базе данных. На картинке сверху справа первые шесть иконок представляют собой привычный комплект из разряда копировать/вырезать/вставить. Тем не менее, обратите внимание на первую иконку (маленькая дискета с черным треугольником). Нажатие этой кнопки сохраняет сделанные записи, поэтому будет неплохо, если вы возьмете за правило сохранять изменения при каждом объемном вводе данных. Обратите внимание, что каждый раз, когда вы доходите до конца строки и внизу появляется чистая, изменения автоматически сохраняются в базе. Но этого не происходит при исправлении уже заполненных строк.

Следующая по-настоящему интересная иконка – это значок «лупа», который скрывает за собой диалог поиска. В нем на данный момент не много пользы, так как наша БД ничтожно мала, но со временем, когда база данных будет расти, эта функция нам еще пригодится. Можно искать текст в полях, текст внутри одного поля, поле целиком, избирательно начало или конец поля, и так далее. Как и во всех других поисковых системах, введите искомое, нажмите кнопку Поиск и откиньтесь на спинку кресла в ожидании результатов!



› Фильтрование позволяет выделить определенные части таблицы. Боже мой, сколько же я всего накупил!



› Сортировка изменяет порядок отображения полей.

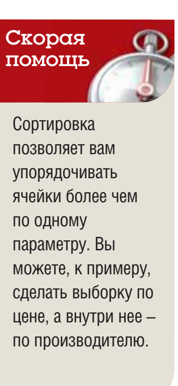
За поиском следует кнопка Обновить/Перечитать, о которой я расскажу в следующий раз. Далее у нас имеются три инструмента сортировки. Первый называется незамысловато: Сортировка. По щелчку появится диалоговое окно, в котором мы можем, к примеру, упорядочить все наши записи по стоимости в виде списка (выберите Возрастающий для того, чтобы в начале были самые дешевые предметы, или букву А и Убывающий для алфавитного просмотра самых дорогих; можно наоборот) или по дате приобретения. Функция сортировки работает по тому же принципу, что и диалог поиска. Задайте образец искомого, и ваши данные будут упорядочены в соответствии с вашим запросом. Если хотите, можно сделать все немного проще – для обычной сортировки по одному параметру, например, щелкните по имени соответствующего поля, а затем по нужной кнопке сортировки (По убыванию или По возрастанию). Вы всегда можете вернуться к оригинальному расположению записей, щелкнув по полю ID и кнопке По Возрастанию.

Отфильтрованное

За вариантами сортировки следуют фильтры. С их помощью мы можем выделить записи или группы записей на основе заданного критерия. Первая иконка называется Быстрый фильтр, в качестве критерия она использует содержимое текущей ячейки. Если я выделю ячейку с ценой £65 и нажму кнопку быстрого фильтра, то все ячейки данного поля, которые не содержат «£65», пропадут, а те, в которых эта цена имеется, поднимутся к верху таблицы. Чтобы сбросить фильтрование, используйте кнопку Сбросить фильтр/сортировку (четвертая в этой группе). Третья по счету иконка вызывает стандартный диалог фильтра, позволяющий задать более гибкие параметры выборки по нескольким полям (похоже на описанный выше диалог поиска).

В нижней части окна редактора таблиц расположены области уведомления и навигации. Их польза проявляется при работе с большими базами данных, когда пользователю важно знать, в какой записи он в данный момент находится и сколько их всего. Здесь также можно удобно перемещаться между отдельными записями при помощи поля Запись x из x и переходить к первой и последней строке таблицы.

Ну вот, наша база данных готова. Как поживает ваша самооценка? Заполнение БД данными может быть скучным занятием, но как только все настроено и готово к использованию, вы можете без труда добавлять новые элементы (например, купленные вещи) в уже созданную структуру. Постаравшись некоторое время, вы получите ощутимое



Скорая помощь

Сортировка позволяет вам упорядочивать ячейки более чем по одному параметру. Вы можете, к примеру, сделать выборку по цене, а внутри нее – по производителю.

» **Через месяц** Мы заполним нашу базу данными из других источников, научимся создавать запросы и отчеты.

Inkscape: создаем



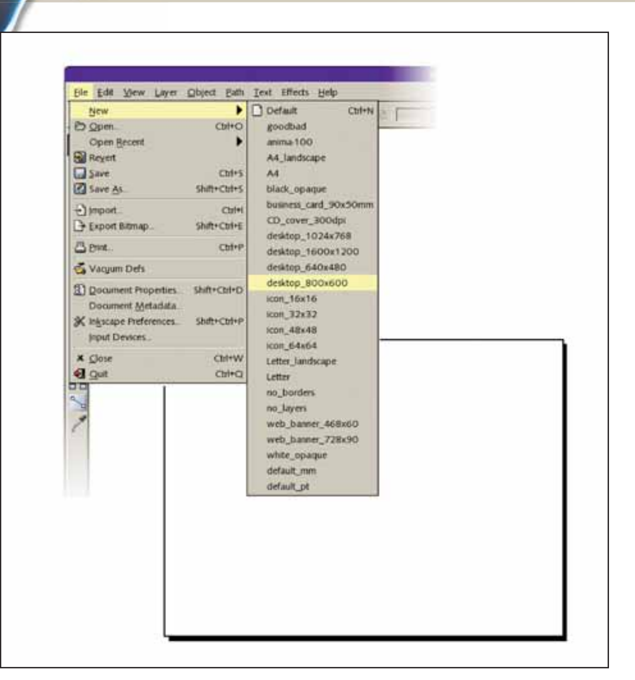
ЧАСТЬ 4: Интернет ждет вас! **Дмитрий Кирсанов** продемонстрирует, на что способен *Inkscape* в области создания web-страниц.

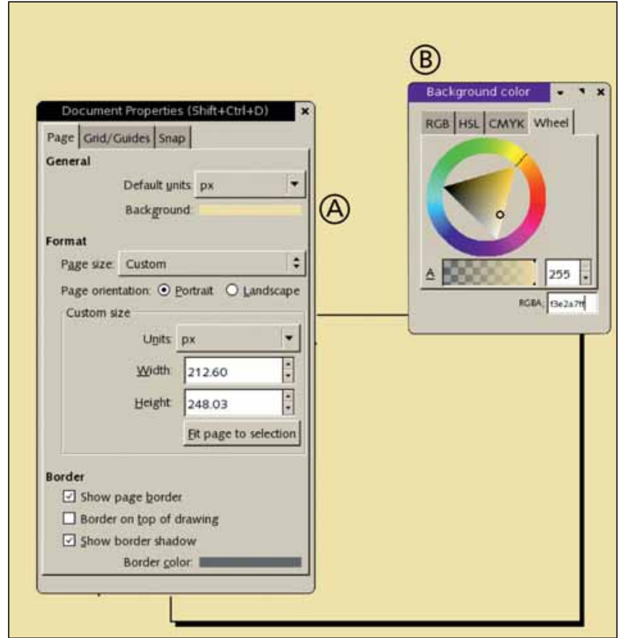
Вспомните время Web 1.0. Dial-up был быстр, #000000 – свеж, а Bangalore был… просто названием города. С тех пор мы пережили бум сайтов типа MySpace, Google Pages и LiveJournal, через которые web-литераторы могут загружать контент в Интернет, не зная ничего о HTML, CSS, не имея даже базовых дизайнерских навыков. Но если вы приложите некоторые усилия, то *Inkscape* поможет вам в создании стильных, уникальных сайтов, созданных специально под ваши цели. Ни в какой программе, кроме векторного редактора, вы не сможете манипулировать объектами – текстом, графикой, фото – с такой свободой и неограниченной гибкостью.

Наш эксперт

Дмитрий Кирсанов
Дизайнер, специализирующийся на создании логотипов и web-сайтов. Консультант, пишет книги и статьи о дизайне и XML-технологиях.

Шаг за шагом: Web-страница за несколько часов





1 Выберите шаблон

Первый шаг, как всегда – создание нового документа по некоему шаблону. В наши дни минимальный разумный размер экрана составляет 800x600 пикселей. Поэтому создание страницы обычно начинают именно с этого разрешения, а уж затем думают о том, как (и если) можно будет развернуть ее на больший экран. Начнем с выбора шаблона под названием Desktop 800x600 в меню File > New (Файл > Создать).

2 Создайте настроение

Цвет – невероятно мощное средство. Страх перед пустым листом (писательский комплекс) намного легче преодолеть, если бумага не чисто белая. Удачно выбранный фоновый цвет задает настроение всему проекту и значительно упрощает выбор остальных элементов. Откройте диалоговое окно Document Preferences (Параметры документа), (Ctrl+Shift+D), выберите Background color (цвет фона) (A) и воспользуйтесь селектором цветов (B). Я выбрал легкий, теплый желтовато-коричневый.

Месяц назад Web-анимация вдохнула жизнь в нашего «тростникового» человечка.

web-страницу

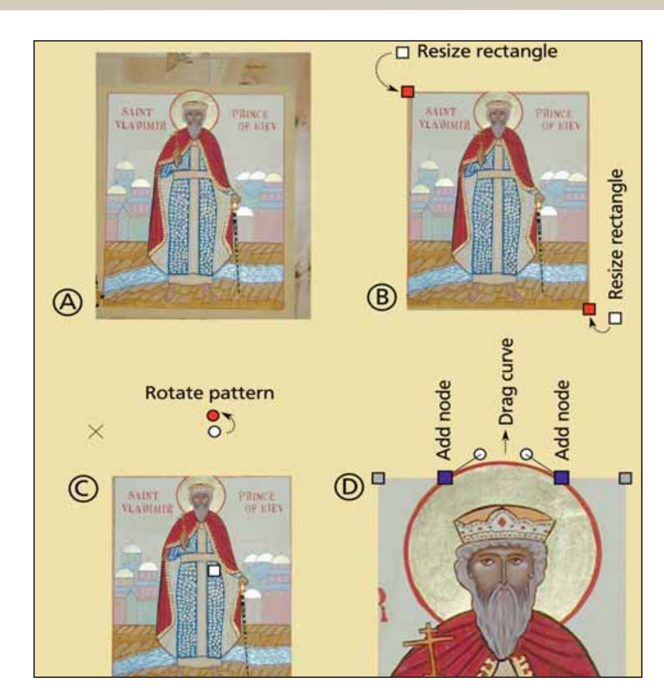
Поэтому ни один другой инструмент не может позволить вам строить вашу страницу с такой точностью и вниманием к деталям. Любая другая программа просто меркнет в сравнении с *Inkscape*.

Я не собираюсь обсуждать использование формата SVG в Интернет непосредственно (хотя такая необходимость давно назрела). Это – тема отдельного разговора. Пока мы будем пользоваться *Inkscape* только для наброски черновика страницы и создания всех ее графических элементов. После этого, настоящую HTML-страницу нужно будет создать отдельно, любым удобным для вас способом. Я не буду подробно освещать эту стадию, но дам несколько советов по ее упрощению.

В качестве практического примера к этой статье я использовал web-сайт, созданный мной для Православной Церкви Св. Владимира, Галифакс, Канада. Мы изучим способы работы с фото, текстовыми колонтитулами и некоторыми другими графическими элементами для сборки их в сбалансированную композицию. Контент сайта очень прост – одна линейная текстовая область, заголовок, изредка – иллюстрация. В этой простоте скрыт некоторый вызов дизайнеру: простому контенту требуется простая структура, но сайт не должен быть безлик или скучен. Разумеется, церковному сайту не пристало быть чересчур крикливым и броским, но примитивности и занудства допускать тоже нельзя. В идеале он должен быть приветливым и душевным, в соответствии с традициями Восточной Православной церкви.

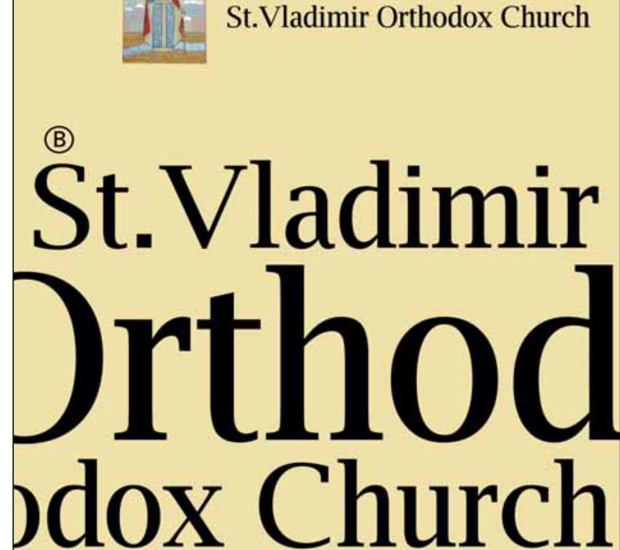
Композиция направлений

Простота этого сайта ставит дизайнеру интересную задачу: как скомпоновать его страницы, чтобы они не выглядели скучными или безликими? Дизайн строится на контрасте и балансе горизонтальных и вертикальных ориентаций. Так как выровненная слева текстовая колонка дает нам вертикальную линию, заголовку нужно придать как можно больше горизонтальных элементов. В основном, контрастные цветные границы горизонтальные, а вертикальные края верхнего и нижнего колонтитулов выведены за пределы страницы. Притапливание вертикальных рамок рисунков также способствует общей горизонтальной ориентации заголовка.



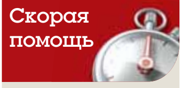
3 Подготовьте ваши рисунки

Займемся фотографией иконы Св. Владимира, святого покровителя прихода. Сначала импортируйте фото в Inkscape через меню File > Import (Файл > Импорт) (A). Выделите его и конвертируйте из объекта-изображения в прямоугольник с заливкой нажатием Ctrl+Y. Можно двигать рукоятки изменения размеров прямоугольника инструментом Node (Узел) (B), поворачивать рисунок-заливку внутри прямоугольника для выравнивания сторон (C) или конвертировать прямоугольник в линии и редактировать его при помощи узлов так, как вам заблагорассудится (D).



4 Выберите подходящий шрифт

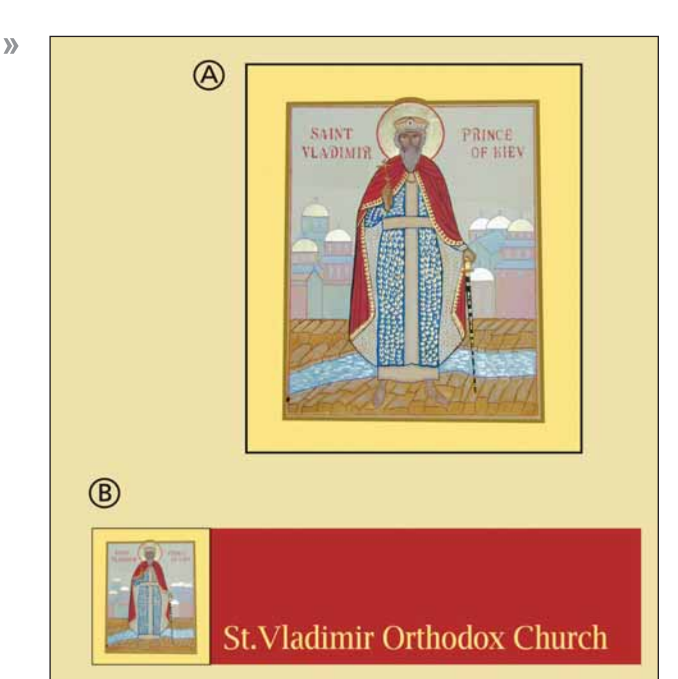
Для простой web-страницы основная часть дизайнерской работы сосредоточивается на ее верхней части, которая обычно состоит из логотипа, заголовка и меню. В нашем случае «логотип» – это икона Св. Владимира, а заголовок – «St.Vladimir Orthodox Church». Сначала поместим их рядом (A). Для заголовка страницы я выбрал Luxi Serif. Этот шрифт выглядит самобытно, и в то же время достаточно традиционно для нашего случая (B) – sans serif был бы чересчур современен. »

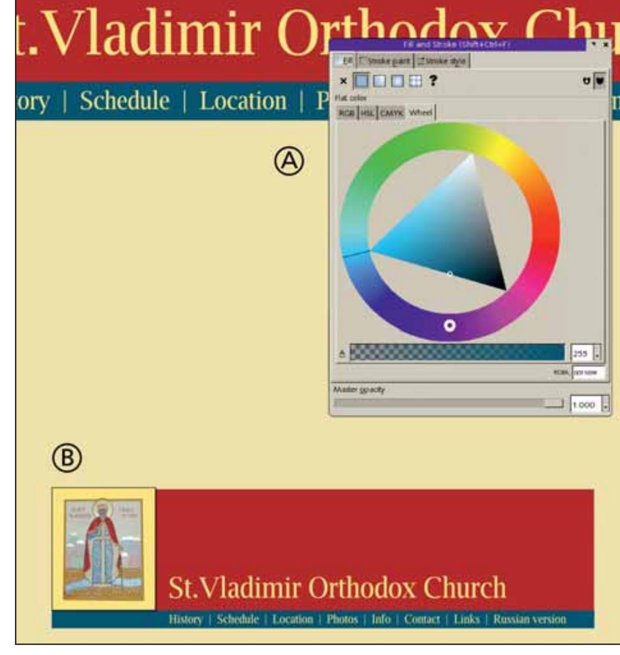


Скорая помощь

Inkscape запоминает разрешение и имя файла каждого импортированного объекта, поэтому ввести имя файла достаточно лишь однажды. Если в дальнейшем понадобится экспортировать некоторые места заново, *Inkscape* сам подставит соответствующее имя в диалоговом окне экспорта.

»



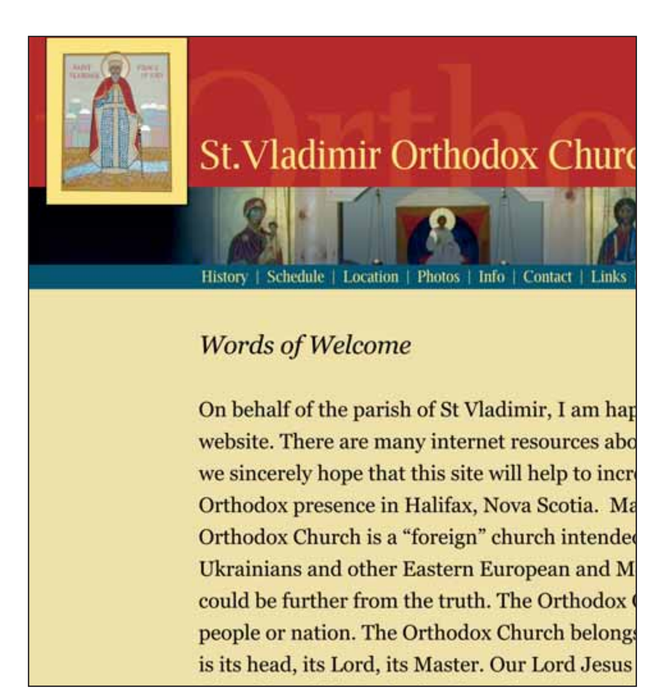


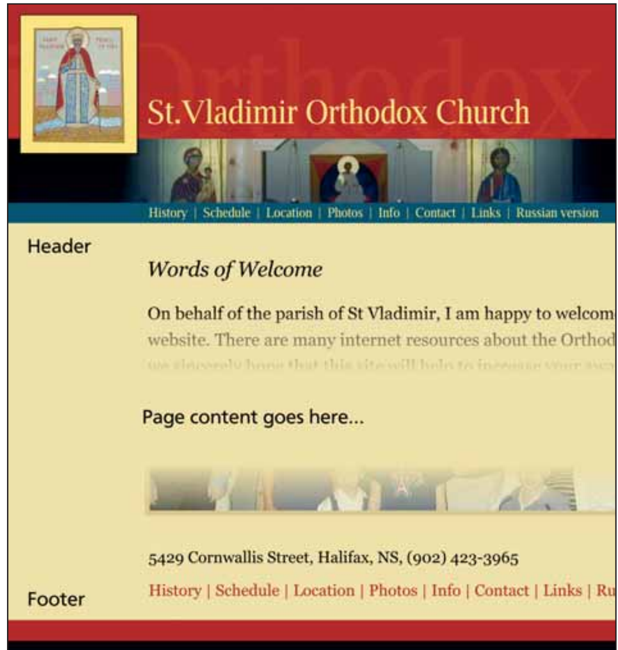
5 Оформите рамку

Православные иконы часто обрамляют роскошными окладами из драгоценных металлов с включением драгоценных камней. Я заключил икону в простой золотистый прямоугольник с узкой черной границей (А). Вполне естественно взять тот же цвет для заголовка, вот только фон потребуется другой. Попробуем темно-красный парчовый оттенок, протянем прямоугольник вправо от иконы и разместим заголовок ближе к нижнему левому углу (B).

6 Организуйте меню

Несмотря на то, что в темно-красном прямоугольнике осталась уйма свободного места, размещать там пункты меню мы не станем – как бы вы ни старались, это будет выглядеть свалкой. Поэтому попробуем добавить ниже композиции контрастную темную зеленовато-синюю полоску (A). Шрифт и цвет для пунктов меню выберем тот же, что и для заголовка. С элементом контрастного цвета наша композиция приобрела стабильность и полноту (B).





9 Оживите страницу

До сих пор наша композиция была совершенно плоской, даже икона не добавила глубины. (Исходя из догматических соображений и традиций, православные иконы не имеют 3D-перспективы вообще или используют обратную перспективу, в которой линии сходятся в сторону наблюдателя, а не в некую отдаленную точку). Давайте добавим еще один элемент (фотографию интерьера церкви) выше полоски меню, затенив ее края полупрозрачными градиентами.

10 Измените заголовок

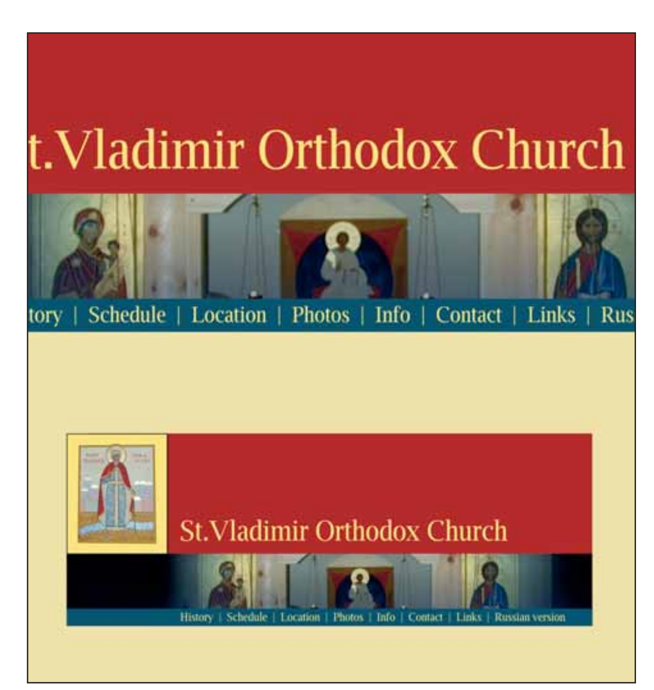
Синеватый градиент и глубина фотографии контрастируют с двумя плоскими пространствами с обеих сторон. Это можно исправить. Сначала сдвинем икону немного вниз и вправо, с тем, чтобы она захватила оба пространства – это сразу оживит композицию (A). Затем попробуем ненавязчиво усложнить фон, скопировав заголовок, увеличив его до 400% и уменьшив яркость до 10% (B).



Скорая помощь

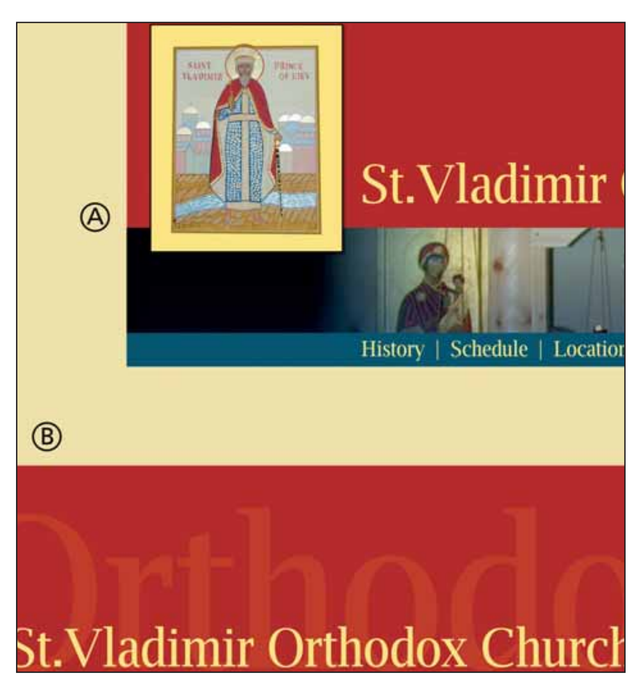
Для экспорта всех объектов из слоя Export можно написать сценарий оболочки (пакетный файл). Для этого необходимо указать значения ID-атрибутов экспортируемых прямоугольников в параметре **--export-id** *Inkscape*. Более подробно об этом см. **man inkscape**.

» Готовый сайт можно просмотреть по адресу: http://users.churchserve.com/ns/ocastvladimir



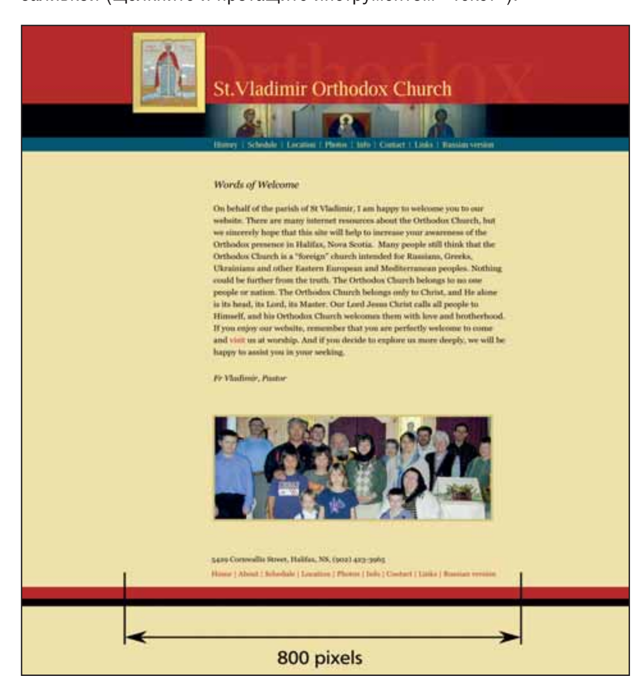
7 Не забудьте про контент!

Пришло время заняться телом страницы. На главной странице должно быть приветственное сообщение, заголовок, подпись и групповое фото общины. Вполне естественно выровнять левый край текста по границе заголовка и меню, оставив поля с левой стороны. Я выбрал шрифт Georgia, поскольку он хорошо подходит к Luxi Serif, использованному в заголовке. Для названий разделов используется *курсив* Georgia. Чтобы создать текстовую колонку, воспользуйтесь текстовой заливкой (щелкните и протащите инструментом «Текст»).



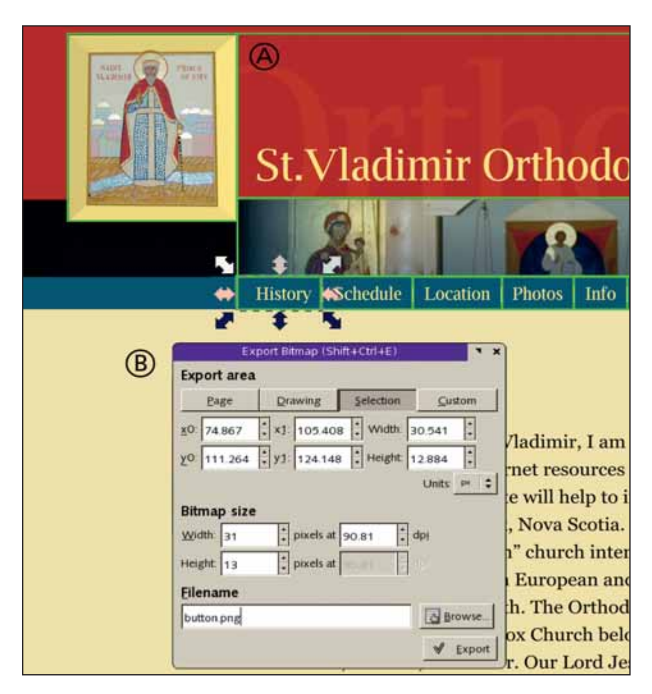
8 Нижний колонтитул

Нам необходимо чем-то завершить каждую страницу снизу. Ничего особенного, просто сбалансируем верхний колонтитул. Двух узких горизонтальных полосок, черной и другого цвета из верхней части, темно-красного, будет достаточно. Даже если верхнее меню текстовое, а не графическое, добрая традиция требует его повторения внизу страницы. Для ссылок вполне разумно использовать тот же «парчовый» цвет, что и в верхнем колонтитуле.



11 Сделайте шаг назад

Теперь самое время отодвинуться и осмотреть страницу в целом. Заодно попробовать, как страница будет вести себя – а значит, выглядеть – на экране размером больше 800 пикселей. Естественное решение – продолжить горизонтальные границы верхнего и нижнего колонтитула до краев экрана, оставив пространство контента (включая левое поле) посередине.



12 Заключительный шаг: Экспорт страницы

К несчастью, пока нет возможности конвертировать SVG-документ в HTML-страницу за один прием. Необходимо будет «разрезать» и экспортировать все графические части нашей страницы – икону, колонтитулы, кнопки – в качестве растровых изображений. Для этого создадим новый слой, назовем его Export, и прочертим невидимые прямоугольники (без заливки, без штрихов) вокруг всех экспортируемых элементов. На нашем рисунке они имеют вид зеленых рамок (A). После этого нужно выделить и экспортировать каждый прямоугольник с помощью диалогового окна экспорта (B). **LXF**

Скорая помощь

Если вы хотите, чтобы экспортированная графика абсолютно, точка в точку, соответствовала размеру экрана в *Inkscape* при масштабе 100%, экспортируемые прямоугольники нужно выровнять по пиксельной сетке. Для ее активации нажмите *#*.

» Через месяц Мы займемся изучением нововведений в *Inkscape 0.44*.

Автоматизируем OpenOffice.org Последнее руководство из серии о макросах

OOo Basic: Используем

ЧАСТЬ 5: Как ни хорош *OOo*, он не распознает макросы Excel VBA, не так ли? Не так. Эксцентричный **Марк Бейн** вам все объяснит.





Наш эксперт

Марк Бэйн
Свою первую программу Марк написал на Sinclair ZX81. Он уже многие годы работает с системами Unix и Oracle, а также читает лекции и пишет.

Мы подходим к концу данной серии уроков, и вы уже знаете, что можно автоматизировать любые задачи с помощью макросов *OpenOffice.org* Basic. На этот раз мы рассмотрим возможность использования Windows-эквивалента, VBA, в *OOo* – и увидим, что не все *OpenOffice.org* одинаковы.

Позвольте мне начать с истории в стиле Макса Байгрейвса [Max Bygraves]. Недавно я купил очень хороший поддержанный ПК, MSI Mega, с Windows XP. Я воздержался от установки на него Linux, и попытался найти способ запустить Linux через Windows. Скажете, кощунство? Объясняю: у компьютера имеется TV-карта, а у меня – странное ощущение, что я останусь без *Freeview*, если мигрирую. Поэтому я установил *Cygwin*, а затем использовал *Cygserver*, чтобы работать с Linux на одной из других моих машин. Но на свете столько дистрибутивов, что я не утерпел, и миграция свершилась. Теперь у меня замечательный ПК с PCLinuxOS, а заодно и ежевечернее развлечение: попытки заставить эту карту снова заработать.

К чему я это рассказал? Очень просто: мой страх потерять *Freeview* аналогичен страху разработчика VBA, который обдумывает миграцию на *OpenOffice.org*. Не станут ли макросы, нажитые непосильным трудом, годны только в помойку, и не придется ли писать все сначала?

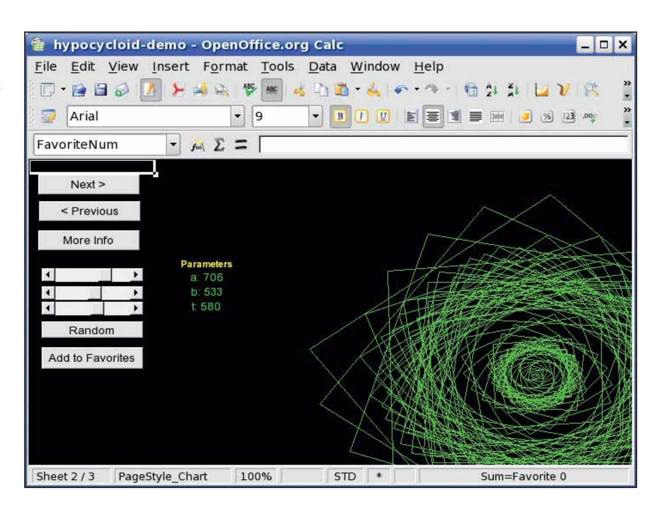
Камень преткновения в Excel

Начнем с простого. Если вы попытаетесь открыть таблицу Microsoft Excel в *OOo Calc*, то все пройдет гладко – *Calc* откроет ее без вопросов, ибо *OpenOffice.org* не страдает предрассудками. А ну как в таблицу встроен макрос? *OOo* выдаст вам стандартное предупреждение «Этот документ содержит макросы», но что бы вы ни выбрали – Включить макросы или Отключить макросы – *Calc* не выдаст никаких ошибок и загрузит документ (**Рис. 1**). Значит ли это, что *OpenOffice.org* понимает макросы VBA? Увы, нет.

Так что же случилось с теми макросами? Открыв свежеимпортированный модуль в редакторе *OOo Basic*, вы обнаружите, почему *OOo* не ругается на код: все строки кода превратились в комментарии. Вот так *OOo* обрабатывает код VBA: просто его игнорирует, предоставляя вам самим во всем разбираться.

Итак, похоже, что страхи разработчика VBA обоснованы – и что данная статья закончится, не начавшись. Я тоже так думал, пока не поговорил с Ноэлем Пауэром [Noel Power]. Этого человека стоит знать – он отвечает за взаимодействие *OpenOffice.org* и Excel VBA в Novell. Ноэль предложил простое решение: установить SUSE 10.1. Давайте подумаем: а надо ли? Какой-то тип из Novell всем рекомендует брать SUSE. С чего это вдруг? «Нет, серьезно, – отвечает Ноэль. – Просто я не знаю, у какого ещё дистрибутива есть поддержка VBA, а у SUSE точно есть.» [Русскоязычные пользователи могут воспользоваться сборками *OOo* от «Инфра-Ресурс» – они поддерживают макросы VBA – прим. ред.] И тут он прав: я-то всегда предполагал (наверно, по наивности), что ваш *OpenOffice.org 2.0.2* на машине с PCLinuxOS (к примеру) и *OpenOffice.org* на машине с SUSE – совершенно одинаковые *OpenOffice.org*. Как бы не так!

Почему же Novell включила поддержку Excel VBA в свою версию *OpenOffice.org*? Потому, говорит Ноэль, «что одним из барьеров внедрения *OpenOffice.org* является отсутствие прозрачной работы макросов. В бизнесе почти все жизненно важные макросы сосредоточены в таблицах *Excel*. Устранение или снижение этого барьера сразу же облегчит внедрение *OpenOffice.org*.» Исследования Novell по использованию VBA на предприятиях показали, что большинство макросов исполь-



(Рис. 2) Файл hypocycloid-demo.xls с разными шаблонами. Calc запросто его откроет, но тут-то и начнутся проблемы...

Месяц назад 14 моих советов по эффективному написанию макросов в OOo Basic.

макросы VBA

LINUX FORMAT на DVD
» hypocycloid-demo.xls
» Пример кода макроса

зуют подмножество объектов Excel API – например, объекты Range и Worksheet. Ноэль и его коллеги сосредоточились на поддержке часто используемых объектов, их методов и свойств, то есть модель Novell VBA хотя и не полна, но покрывает большую часть потребностей.

SUSE как спирограф

Пришла пора установить SUSE 10.1. Должен признаться, что SUSE 7.1 был моим первым дистрибутивом Linux, когда я начал работать на себя в начале этого столетия. К сожалению, я знал, что SUSE хоть и великая вещь, но чересчур великая для моего тогдашнего ноутбука. Однако ПК с тех пор несчетно размножились, поэтому хорошо, что есть повод вернуться к SUSE. А как с установкой? Да легко.

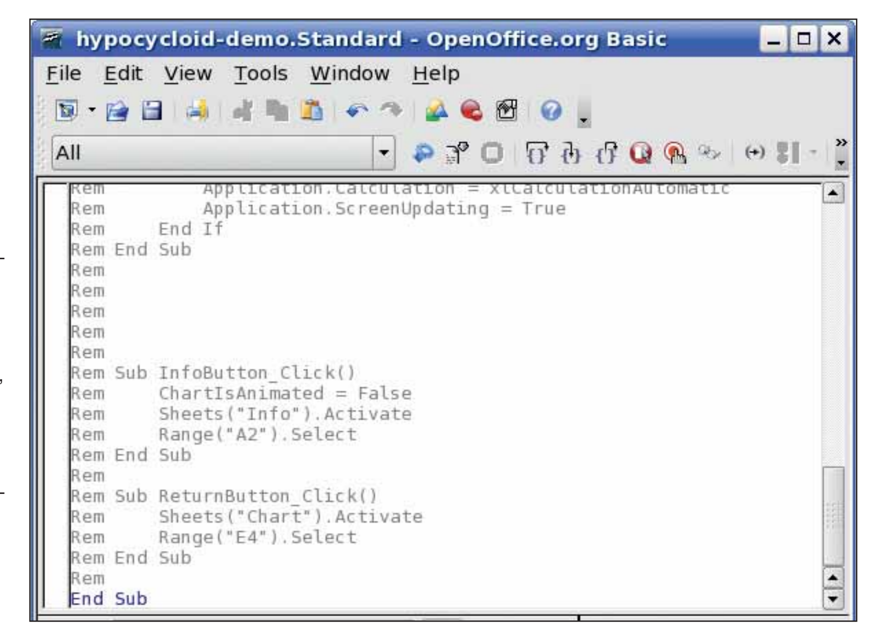
Ну, на самом деле не совсем. Я скачал 5 образов CD с сайта Novell, записал их на диски, перегрузился и понял, что создал 5 подставок для пивных кружек. Поделюсь советом: когда на диске написано, что скорость прожига 52x – не верьте. Лучше сбросьте скорость записи до 12x – на запись потребуется больше времени, но зато она будет сделана качественно. Что до установки – легко. Скачал 5 образов CD с сайта Novell, записал их на скорости 12x, перегрузись и следуй инструкциям инсталлятора.

Если вы сделаете то же самое, перед вами предстанет такая же копия *OpenOffice.org*, как и везде. Существующие пользователи SUSE 10.1 могут даже не понимать, из-за чего тут суетиться. Конечно, если вам не нужны макросы Excel VBA, то вы всего лишь поимели новую версию *OpenOffice.org*. А вот если к вам попала таблица *Excel* с макросом, придется побеспокоиться. Каков же следующий шаг? Загрузим табличку Excel (естественно, содержащую макрос) и посмотрим, что произойдет.

Вы захотите сразу же увидеть рабочий пример, поэтому мы записали на диск файл **hypocycloid-demo.xls** от Ноэля. Что в нем особенного? Ничего. Вы его загружаете, а он работает: нажмите на кнопку – и картинка изменится (**Рис. 2**). Все здорово, но реальную разницу вы увидите, только просмотрев макрос в редакторе кода. Для этого надо прощёлкать Tools–>Macros–>Organize Macros–>OpenOffice.org Basic. На экране OpenOffice.org Macros разверните Hypocycloid-demo, затем Standard. Наконец, нажмите Module1 и Edit.

С первого взгляда вы не увидите ничего необычного (**Рис. 3**). Но вспомните, как вы открывали макрос в не-SUSE версии *OpenOffice.org*. На сей раз весь код, строчка за строчкой, целехонек; велите ему работать, он и будет работать.

Это означает две вещи. Первое: если вы работаете на Excel VBA, при миграции на *OpenOffice.org* весь свой код выбрасывать вам не придется. Однако не забудьте сохранить свои наработки, прежде чем окончательно избавиться от Windows. Второе, не менее важное: можно продолжать разработку со старыми знаниями, просто используя *OpenOffice.org* вместо *Excel*. Ура!



› **(Рис. 1) Код VBA не вызовет затруднений – OOo просто его закомментирует.**

На практике

Не надо песен: вы пошли и загрузили парочку ваших старых макросов *Excel*, они идеально работали, а когда вы попытались создать макрос для новой таблицы, то потерпели неудачу. Например, вы могли бы попытаться написать следующее:

```
Sub writeToCell_VBA
        sheets("Sheet1").select
        cells(1,1).value="Hello world"
End Sub
```

Вы, возможно, получаете ошибку 'BASIC runtime error. Sub-procedure or function procedure not defined' Не беспокойтесь. Все, что вам надо сделать, это добавить дополнительный параметр в начало каждого модуля, в котором вы хотите использовать VBA:

```
Option VBASupport 1
```

Вот так и поступает Novell-версия *OOo* – не превращает строки кода в комментарии, а вставляет **Option VBASupport** 1 в начале модуля (**Рис. 4**).

Здесь возникает вопрос: только ли это поддержка VBA? В конце концов, кроме упрощения возможности миграции ваших *Excel*-макросов в *OpenOffice.org*, вы получили новый метод программирования, который вы можете найти более удобным в использовании, чем собственно *OOo* Basic. В этом и прелесть Linux: чтобы выполнить работу, нет правильного и неправильного пути. Давайте рассмотрим процедуру записи в ячейку при помощи *OOo* Basic:

```
Sub writeToCell_OOo
        dim sheet as object
```

»

Что такое гипоциклоида?

Протестировав файл, вы можете заинтересоваться: что такое гипоциклоида? Вот краткий урок математики:

Циклоиды – класс кривых, по которым движется точка на радиусе круга, катящегося без скольжения по некой направляющей линии.

Если эта направляющая – прямая, получается трохоида. Если это внешняя сторона другого круга – эпициклоида, а если внутренняя – гипоциклоида.

Тут вы воскликнете: «Так это же спирограф!» – и сразу станет ясно, сколько вам лет…



```
dim cell as object
 sheet=thisComponent.sheets(0)
 cell=sheet.getCellByPosition(0,0)
 cell.string="Hello World Again"
End Sub
```

Вы можете сократить этот код до

```
Sub writeToCell_minimum
 thisComponent.sheets(0).
getCellByPosition(0,1).
 string="Last One"
End Sub
```

На этих примерах вы, возможно, разглядите одно из больших отличий *OOo* Basic и VBA. *OpenOffice.org* Basic содержит главный объект (**thisComponent**), и все наследуется от него. VBA работает с отдельными объектами (**Application**, **ActiveSheet**, **ActiveCell** и им подобными). Их использование может сделать код читабельнее и проще в поддержке.

В некоторых случаях вы обнаружите, что *OOo* Basic и VBA делают одно и то же, просто используя разные слова. Например, взгляните на:

```
ThisComponent.LockControllers 'формат OOo
```

и

```
Application.ScreenUpdating = False 'формат VBA
```

Обе этих строки выполняют одно и то же: замораживают экран, это может быть полезно, если вы пишете в несколько ячеек. Чем сидеть и смотреть, как поштучно происходит обновление ячеек, лучше увидать полностью готовый экран, когда работа выполнена. И не забудьте разморозить экран:

```
ThisComponent.UnlockControllers 'формат OOo
```

или

```
Application.ScreenUpdating = True 'формат VBA
```

Большинство случаев похожи на этот: та же функциональность, но все слегка упрощено благодаря новым объектам. Мы уже видели, как работать с ячейками; а сейчас мы покажем, как устанавливать диапазон значений в VBA и *OpenOffice.org*:

```
ThisComponent.CurrentController.ActiveSheet.
    getCellRangeByName("B2").String = _
"Используем Windows, когда так просто установить Linux"
```

и

```
Range("B1").Value = "Что такое мазохизм"
```

Если вы читали внимательно, то заметили, как вводится информация в ячейки и диапазоны. В формате *OOo* Basic вы просто указываете тип данных, которые собираетесь загрузить в ячейки (строка или значение). Если вы используете формат VBA, у вас только один тип данных: значение. Разница невелика, но знать о ней надо.

Как мы уже видели, большая часть функциональности VBA реализуется с помощью функций простого формата. Иногда он совсем прост. Например, чтобы очистить диапазон ячеек, вы можете использовать либо

```
Range("A1:E5").ClearContents
```

либо

```
ThisComponent.Sheets.getByName("Sheet1").
getCellRangeByName("B1:E5").clearContents( _
com.sun.star.sheet.CellFlags.VALUE _
+ com.sun.star.sheet.CellFlags.STRING _
+ com.sun.star.sheet.CellFlags.DATETIME _
+ com.sun.star.sheet.CellFlags.ANNOTATION _
+ com.sun.star.sheet.CellFlags.FORMULA _
+ com.sun.star.sheet.CellFlags.HARDATTR _
+ com.sun.star.sheet.CellFlags.STYLES _
+ com.sun.star.sheet.CellFlags.OBJECTS _
+ com.sun.star.sheet.CellFlags.EDITATTR)
```

Выбор остается за вами – бывают ведь и любители побарабанить по клавишам. Знаю, знаю, можно вместо этого написать процедуру, но чего ради, если работа уже сделана за вас?

Скорая помощь

OOo безразличен к регистру (вне зависимости от поддержки Excel VBA). Однако постарайтесь привыкнуть к использованию заглавных букв, чтобы сделать код более читабельным, например, пишите openNewSheet вместо opennewsheet. Зачем? А чтоб уменьшить вероятность сделать ошибку.

```
    Sheets("Info").Activate
    Range("A2").Select
End Sub

Sub ReturnButton_Click()
    Sheets("Chart").Activate
    Range("E4").Select
End Sub
```

(Рис. 3) На этот раз Calc не закомментировал код VBA, а принял его.

Испытайте на своем дистрибутиве

Короче, вы поняли, в чем штука. Если вы пишете на VBA, то и продолжайте себе писать. Если он для вас в новинку и вам понравился, вы, возможно, захотите узнать о нем побольше. В пакете вы помощи не найдете. Поискав в файлах справки *OpenOffice.org* по ключу VBA, получите информацию о том, как работает обычная редакция *OOo* – это мы уже проходили: код VBA грузится как комментарии. Однако по Excel VBA существует уже так много информации, что даже поиск в Google позволит найти полное руководство к действию.

Отсутствие документации свидетельствует о статусе проекта, а не о его качестве. В конце концов, в любом проекте бумажная работа всегда оставляется на потом.

Всегда следует помнить, что проект находится в стадии развития. Вы можете обнаружить, что какой-то код не импортируется или вызывает ошибки. Однако в основном это проявляется разве что при использовании редких и сложных функций.

Здесь-то вы и можете помочь. Если вы уже используете SUSE 10.1 или планируете на него перейти, я ожидаю, что вы скажете: «А в чем проблема?» Ну и прекрасно. Однако если вы приверженец Debian или Fedora, или любой другой системы, то проблемы будут – ну, то есть, будут, если вам потребуется взаимодействие с Excel VBA. Что вы можете предпринять?

Ноэль сказал мне, что Novell работает над включением модели VBA в исходный код стандартной поставки *OpenOffice.org*, и недавно основал проект продвижения работы Novell по VBA (http://vba.OpenOffice.org) в состав *OOo*, что позволит большему числу пользователей Excel VBA перейти на открытое программное обеспечение. «Наката Махо [Nakata Maho], разработчик *OpenOffice.org*, и Ханно Мейер-Туров [Hanno Meyer-Thurow], разработчик Gentoo, помогли в реализации и разработке API», – сказал он мне. Если вы решили установить модель VBA на вашем дистрибутиве, то найдете исходный код на http://go-oo.org, а инструкции по скачиванию на http://wiki.services.OpenOffice.org/wiki/Getting_It. Однако для экономии времени можно сделать вот что:

```
export CVSROOT=':pserver:anonymous@anoncvs.gnome.org:/
cvs/gnome'
cvs login
cvs -z3 checkout -P OOo-build
```

Вы обнаружите, что у вас создался каталог **OOo-build**. Полные инструкции по сборке расположены по адресу http://wiki.services.OpenOffice.org/wiki/Building. Процесс достаточно прямолинеен, а если и сложноват, то полнота описания снижает вероятность совершения ошибки. Так как **OOo-build** – действительно разрабатываемая версия,



› (Рис. 4) Установите VBASupport 1, чтобы избежать этой ошибки.

то вы можете обнаружить, что придется добавить другие приложения – но если что-то потребуется, **OOo** вам об этом сообщит.

Поэтому на следующем шаге вы идете в каталог **OOo-build** и запускаете **autogen.sh**. На моем Debian он не заработал из-за ошибки './autogen.sh: line 18:aclocal:command not found'. Быстрое прочесывание Интернета показало, что лекарство – установка *automake*:

```
sudo apt-get install automake
```

Когда вы заставите работать *autogen.sh*, следующим шагом будет выбор дистрибутива, на котором планируется установка. Выбирайте из Ark, Ark64, Common, CommonLinux, CommonWin32, Debian, Debian64, DebianSarge, DroplineGNOME, Frugalware, Frugalware64, Gentoo, Mandriva, Mandriva64, NovellWin32, QiLinux, SUSE-10.0, SUSE-10.1, SUSE-64. SUSE9.1, SUSE-9.2, SUSE-9.3, SUSE-PPC, SUSE, Ubuntu, Ubuntu64, UbuntuL10n, translate-org-za-Linux и translate-org-za-win32. Все что вам надо сделать, это запустить **autogen.sh** с опцией **--with-distro**:

```
/autogen.sh --with-distro=DebianSarge
```

Далее следим за работой *autogen.sh* и сообщениями об ошибках, а затем устанавливаем недостающие программы:

```
autogen.sh stops with the error:
Install the software:
configure: error: install pam-devel
```

> «При переходе на OOo не придется выкидывать вон свои наработки.»

```
sudo apt-get install libpam0g-dev
configure: error: install png-devel
sudo apt-get install libpng2-dev
```

Чтобы отыскать все программы, придется пошарить в Интернете, но на базовую конфигурацию много времени не уйдет.

Далее нужно заставить **OOo-build** скачать необходимые файлы, это сделать легко: просто запустите **./download**. Наконец, запустите **make**. Ну, не совсем наконец, потому что вы можете получить следующее сообщение об ошибке:

```
bainm@hector:~/OOo-build$ make
Making all in po
make[1]: Entering directory `/home/bainm/OOo-build/po'
make[1]: *** No rule to make target `yes', needed by `all-am'.
Stop.
make[1]: Leaving directory `/home/bainm/OOo-build/po'
make: *** [all-recursive] Error 1
```

Присказка наших дней – «В беде обращайтесь к Google» (если, конечно, вы не в Китае). Я поискал и обнаружил, что у других людей наблюдается та же проблема. К счастью, решение простое: все Makefile содержат строку **all-am: Makefile $(SCRIPTS)**, тогда как она должна быть **all-am: Makefile**. Чтобы это исправить, надо отредактировать около 25 Makefile'ов – либо написать простой скрипт:

```
FILES="Makefile */Makefile */*/Makefile"
for FILE in $FILES
do
cat $FILE | sed s/"all-am: Makefile \$(SCRIPTS)"/"all-am:
Makefile"/g > $FILE.tmp
mv $FILE.tmp $FILE
done
```

После этого **make** должен работать правильно. Однако, если у вас проблемы с программами, можете отключить их с **autogen.sh**. Потом можете воспользоваться советом *autogen.sh* – расслабиться и выпить чашку чая – хотя у меня это вылилось в питье чая, прогулку с собакой, кормление кошек и поход в бар.

Вот и все, уважаемый читатель. Агитируйте разработчиков вашего любимого дистрибутива последовать примеру Novell. Примите участие в сообществе взаимодействия с VBA. Или начните использовать новую версию *OpenOffice.org*.

Если вам интересно узнать о вышеупомянутой TV-карте – она заработала, как только я установил SUSE 10.1. Вообще-то я использую Kaffeine для просмотра каналов – он проще в использовании и стабильнее, чем исходная Windows-версия программы. Ну как тут не любить Linux? LXF

Скорая помощь

Если вы собрались писать программы в стиле VBA, то не забывайте, что первой строкой каждого модуля должна быть

```
Option
VBASupport 1
```

В противном случае вы получите синтаксическую ошибку.

Облегчите себе жизнь

Если вы не верите, что взаимодействие *OpenOffice.org* и Excel VBA поможет облегчить вам жизнь, просто гляньте, как легко создать новую таблицу и начать в ней писать:

```
Dim wSheet as worksheet
Dim wBook as workbook
set wBook = application.workbooks.add
set wSheet = Sheets(0)
wSheet.cells(2,1).value = "Too easy for words"
```

Изменить шрифты также просто:

```
wSheet.Range("A1:Z1").Select
With Selection.Font
        .Name ="Verdana"
        .Size = 14
End With
```

А как насчет того, чтобы спрятать один из рабочих листов?

```
Dim wSheet2 as worksheet
set wSheet2 = Sheets(2)
wSheet2.visible = false
```

Как программист-лентяй, я благодарен людям, выполняющим трудную работу по переносу Excel VBA в OpenOffice.org.

Новая серия! Как воспользоваться умными утилитами Linux и защитить вашу машину.

Безопасность:

ЧАСТЬ 1: Первая линия обороны Linux-системы – механизм пользовательских паролей. **Д-р Крис Браун** объяснит, как он работает и как ещё можно укрепить защиту системы.





Наш эксперт

Д-р Крис Браун независимый инструктор по Linux с сертификатами Novell и Red Hat, имеет степень доктора наук по физике элементарных частиц, недавно написал книгу о SUSE для издательства O'Reilly.

«Уверенность», по словам Гекаты в трагедии Шекспира *«Макбет»*, «несет погибель смертным». Она хотела сказать, что чрезмерная вера Макбета в себя в конце концов его погубит: ведь считая себя неуязвимым, он необоснованно рисковал. Прямо как вы! Вообразили, что ваш Linux неуязвим? Вы идете к пропасти…

Есть немало свидетельств, что работать в Linux куда безопаснее, чем в Windows. Я сам видел цифры: Linux «из коробки» подхватывает заразу за три месяца, тогда как Windows – за три часа (см. www.honeynet.org). Но это не повод относиться к безопасности своей системы спустя рукава.

В данной серии, состоящих из пяти уроков по безопасности Linux и соответствующим открытым утилитам, мы рассмотрим, как обнаруживать уязвимости и прикрывать их. В первой части мы рассмотрим пароли и поговорим о контроле над привилегиями, даваемом *sudo*.

Между безопасностью и удобством всегда существует компромисс. Машина, запертая в комнате, где нет подключения к сети, конечно, намного сохраннее, чем машина с доступом в Интернет посреди проходного двора, однако настолько же и бесполезнее. Попробуем облечь убытки от уязвимостей в наукообразный вид – получится такая формула:

Ожидаемые убытки = (вероятность успешного вторжения) x (цена этого вторжения)

На этих уроках я буду говорить об уменьшении вероятности взлома, но важно также уменьшить потери от успешного вторжения. Например, как можно скорее обнаружить вторжение гораздо лучше, чем на много дней оставить его безнаказанным. Регулярное создание резервных копий существенно сократит возможные потери данных. Распределение сервисов на несколько машин (не кладите все яйца в одну корзину) поможет предотвратить простои в работе организации из-за единичного взлома. И так далее.

Хэшируй это!

Начнем с начал: Linux – многопользовательская операционная система. Когда вы входите в нее, вам предложат ввести имя пользователя и пароль для подтверждения того, что вы – это вы. Выбор хорошего пароля – ключевая часть сохранения безопасности системы.

Вот как работает механизм паролей в Linux. Информация о пользовательских учетных записях хранится в двух файлах: **/etc/passwd** и **/etc/shadow**. Файл **passwd** содержит имя пользователя, его числовой идентификатор (User ID, UID) и домашнюю директорию, но (вопреки названию) пароля-то в нем и нет: он укрылся в файле **shadow** (далее мы увидим, зачем нужно разделение на два файла). Файлы **passwd** и **shadow** параллельны – каждой учетной записи пользователя соответствует строка в одном файле и строка в другом. Когда пользователь устанавливает пароль, пароль хэшируется, а результат хэширования помещается в **/etc/shadow**.

Хэширование похоже на шифрование, в том смысле что из внятного текста получается полная белиберда; однако, в отличие от шифрования, хэширование – операция необратимая. Каждый раз, когда вы вводите пароль, он хэшируется, и полученный хэш сравнивается с тем, что прописан в **/etc/shadow**. Linux не хранит пароли в чистом виде.

В зависимости от вашей Linux-системы, у вас есть выбор из трех путей хэширования:

1 Функция crypt от Unix, болтающаяся в системе уже много лет. Команда crypt реализует алгоритм DES (Data Encryption Standard) и генерирует 13-символьный хэш.

2 MD5, который, как вы имели шанс заметить, используется для верификации ISO-образов компакт-дисков и передаваемых по сети файлов, а также в цифровых подписях. MD5 генерирует 34-символьный хэш.

3 Blowfish (или, скорее, хэш-функция, выведенная по алгоритму Blowfish), генерирующий 60-символьный хэш.

Все эти алгоритмы используют «соль» (salt) – случайное число, генерируемое при установке пароля. Соль и указанный пользователем пароль комбинируются и хэшируются. Соль и хэш сохраняются в **/etc/shadow**. Если вы проверите свой **/etc/shadow** на системе, использующей хэши MD5, то ясно разглядите соль и хэш. Например, в строке пароля **1Ue1yZO5i$.6G/4l1AYliPdkTh1AkBi/** восемь символов после **1** – соль, а 22 после последнего **$** – хэш. Мы увидим, что соль делает взлом паролей гораздо труднее.

Отчасти стойкость пароля зависит от качества хэш-алгоритма. Хороший хэш-алгоритм в теории делает невозможным вычисление



Стойкие пароли

пароля по хэшу. Конечно, зная хэш, вы всегда можете попробовать метод грубой силы – хэшировать каждую возможную последовательность из (допустим) восьми символов в надежде найти соответствие. Но существует около 90 отображаемых ASCII-символов, а это значит, что перебрать придется 90 в восьмой степени вариантов – то есть 4,304,672,100,000,000.

Если ваша машина способна вычислять 10000 хэшей в секунду, то на весь перебор понадобится 13650 лет, а это, вероятно, дольше, чем вам хотелось бы ждать. Соль еще более усложняет взлом, делая невозможным простое сравнение со списком хэшей для наиболее часто употребляемых паролей, так что взломщики могут искать совпадение только с одним паролем за раз.

Наш анализ немного наивен, поскольку длина хэша тоже влияет на интервал поиска, но по крайней мере будьте уверены – атаки грубой силой (перебором) слишком медленны. (Более глубокий анализ можно найти в книге Брюса Шнайдера «*Прикладная криптография*», более поверхностный – в «*Цифровой крепости*» Дэна Брауна). К сожалению, реальный интервал поиска гораздо меньше, поскольку многие люди выбирают для пароля простые слова или имена: любимого супруга или собаки, или, того хуже, собственное имя пользователя.

Потрошим колбасу

Программы-взломщики паролей автоматизируют атаки, выбирая из словаря пароль, из которого получается данный хэш. Они пользуются информацией из вашей базы данных паролей, и, увы, добиваются успеха пугающе часто. Возьмем, к примеру, популярную утилиту *John the Ripper* [Джон-Потрошитель, – прим. ред.].

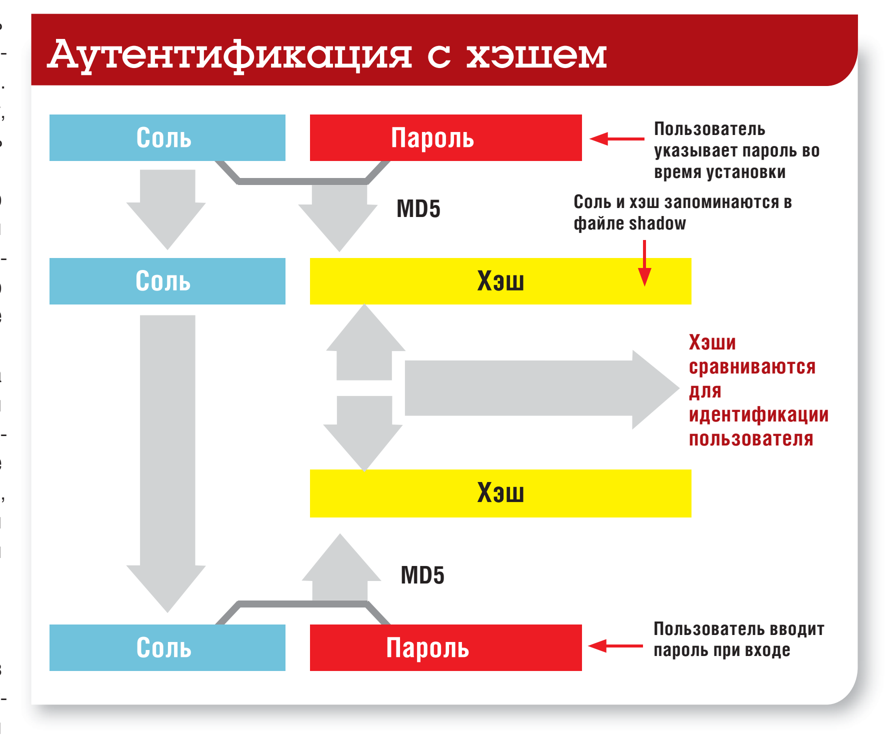
Вы можете скачать «Джона» с **www.openwall.com**. Перед запуском слейте содержимое **/etc/passwd** и **/etc/shadow** в один файл. Это делает команда *unshadow* (включенная в пакет *John the Ripper*):

```
# unshadow /etc/passwd /etc/shadow > passwdcombo
```

В результате получится файл, где хэши будут вставлены во второе поле файла **passwd** (как это было до изобретения shadow). Для выполнения описанной команды вы должны иметь привилегии суперпользователя, поскольку только он может читать **/etc/shadow**. Теперь запускайте *John*. Опций командной строки для настройки поиска хватает, но благодаря разумным установкам по умолчанию *John* будет работать, если ему просто указать имя полученного выше файла:

```
# John passwdcombo
```

Аутентификация с хэшем



Не ждите, что результаты появятся сразу. Для тестирования я создал десять пользовательских учетных записей со следующими паролями: user1pw; secret; terces; secret42; Secret; ducttape; England; pit; coat; milewide; sausages [те самые колбасы, – прим. ред.]. Из них *John* нашел secret, terces и Secret в течение нескольких минут. В течение ночи он нашел England и sausages. Пять из десяти, хотя, конечно, некоторые из этих паролей были нарочито слабыми.

В тесте участвовала версия *John the Ripper 1.6-4* на Fedora Core 5 с MD5-хэшами. Если угодно, настраивается *John* несколькими способами: подключением к нему разных списков паролей; указанием правил по трансформации паролей из списка; через файл **John.ini**; через аргументы командной строки.

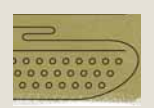
Успех программ-взломщиков паролей поясняет, почему пароли не хранятся в **/etc/passwd**: этот файл могут читать все пользователи (он часто используется для поиска соответствия между пользователем и UID), так что если бы хэши хранились здесь, любой пользователь мог бы начать подбор паролей. Перемещение хэшей в **/etc/shadow** (который может читать только root) блокировало эту линию атаки.

Наши результаты должны также укрепить в вас сознание необходимости выбирать сильные пароли. Те, кто использует слова, имеющиеся в словаре, просто нарываются на неприятности. С другой стороны, использование слишком сложного для запоминания и набора пароля также неблагоразумно. (У одних моих клиентов пароль root был наклеен на терминал).

Один из способов составления хорошего пароля – выбрать два несвязанных слова и склеить их каким-нибудь знаком пунктуации; например так: pico:drip или tiny-duck. Другой способ – взять слово, заменить часть букв на заглавные, а часть – на цифры: B33rMats (beermats), a1rcr4FT (aircraft).

Добиться того, чтобы ваши пользователи использовали сильные пароли – вопрос обучения. Объясните, как выбрать хороший пароль, »

Этика взломщика



John the Ripper – одна из утилит для оценки уязвимости, которыми пользуются и «темные», и «светлые». Первые пользуются ею для взлома паролей. Вторые применяют ее для обнаружения слабых паролей, пока до них не добрались плохие парни. Так что для порядка нужно бы написать про отказ от ответственности. Во-первых, я (как и вся редакция *LinuxFormat*) абсолютно не одобряю использование этой утилиты в корыстных целях. Во-вторых, если вы решили применить эту утилиту на работе, сначала получите «добро» от своего начальника. Есть реальный шанс, что вас уволят, если обнаружится, что вы запускаете взломщик на рабочих машинах без разрешения, даже если вы руководствуетесь благими намерениями.



Установите пароль на Grub для предотвращения доступа через загрузку в однопользовательском режиме



» и регулярно запускайте взломщик, выявляя тех, кто не слушает советов. Если вы хотите автоматизировать принудительный выбор сильных паролей, рассмотрите как вариант установку модуля PAM *pam_passwdqc*. Этому модулю можно поручить установку паролей, а заодно настроить его на проверку паролей по длине и смешиваемости символов (верхнего/нижнего регистра, цифр и других знаков). Модуль доступен на www.openwall.com.

Установка пароля Grub

Взлом пароля – не единственный способ проникнуть в Linux-систему. Большинство систем в качестве загрузчика используют *Grub*. *Grub* достаточно гибок в настройке и включает опции вмешательства в процесс загрузки и редактирования команд загрузки (или даже добавления новых команд).

Grub умеет передавать аргументы ядру Linux, и один особо интересный трюк заключается в добавлении аргумента **init=/bin/sh** в конец команды загрузки ядра. Этот аргумент скажет ядру, что вместо команды *init* нужно запустить **/bin/sh**, и вы получите полный доступ к системе без ввода пароля суперпользователя.

Атаки такого рода можно предотвратить, установив пароль на *Grub*. Тогда интерактивные опции *Grub* не включатся до тех пор, пока не будет введен верный пароль. Пароль нужен не на загрузку согласно имеющемуся файлу конфигурации *Grub*, а на редактирование и интерактивный ввод команд.

Пароль для *Grub* установить легко. Откройте файл **/boot/Grub/menu.lst** и добавьте в глобальные настройки (в начале файла) строку вроде этой:

```
password --md5 $1$H06141$PTIpTGW7fNKspluqd1Mdk.
```

Мы опять видим MD5-хэш (вместе с солью), который мы уже встречали ранее. Есть несколько способов его генерации. Во-первых, можно просто скопировать хэш существующей учетной записи из файла **/etc/shadow**. Естественно, тот, для которого вы знаете пароль. Во-вторых, можно запустить *Grub* из командной строки и ввести команду *md5crypt*. Программа попросит вас ввести пароль и выведет хэш. Диалог выглядит так:

```
# Grub
 GNU Grub version 0.95 (640K lower / 3072K upper memory)
Grub> md5crypt
Password: *********
Encrypted: $1$H06141$PTIpTGW7fNKspluqd1Mdk.
Grub> quit
(Теперь можно вставить этот хэш в /boot/Grub/menu.lst)
```

Если вы теперь загрузите машину, имея эту строку в файле конфигурации *Grub*, то сможете только выбирать ОС из предопределенных в **menu.lst**, клавишами «вверх» и «вниз», а для других действий придется нажать **p** и ввести пароль.

В мультизагрузочной системе можно также заблокировать определенные пункты меню, и доступ к ним получат только знающие пароль, причем пароль у каждого пункта может быть своим. Для этого ознакомьтесь с руководством по *Grub*, доступном на www.gnu.org/software/Grub/manual.

Скорая помощь

Хотя сильные пароли трудно запомнить, но лучше брать такие, чем слово из словаря или имя любимой.

Атаки физического доступа

Перехват процесса загрузки *Grub* требует физического доступа к машине, и если этот доступ есть, то есть и способы обхода пароля. Например, кнопка Reset вашего компьютера заслуживает бирки «Для root-доступа нажимать здесь». Возьмите LiveCD с диска к нашему журналу и загрузитесь с него, да даже просто выберите спасательный (rescue) режим загрузки штатной ОС со стандартного носителя – и в любом из этих случаев у вас будет полный доступ ко всей файловой системе. В конце концов, если и это не удалось, всегда можно взять отвертку и унести жесткий диск в кармане.

Контроль доступа

Не сомневаюсь: вы уже знаете, что в Linux есть специальная учетная запись, называемая суперпользователем. Общепринятое имя для этой записи – root, однако ключевой момент – числовой пользовательский идентификатор, UID: для root он равен нулю. Если вы вошли в систему как root, считайте все проверки пройденными, а все права доступа вашими – вы всемогущи. Как модель безопасности, понятие root вызывает примитивное чувство все-или-ничего.

К примеру, если вы хотите позволить кому-нибудь подстраивать системные часы, дайте ему пароль root, который позволит ему влезть в секреты своего начальника. С точки зрения ролевой системы, по контролю доступа ролей только две: смертный человек и Бог. Для потенциального взломщика получение пароля root на вашей системе – абсолютная победа.

Команда *sudo* позволяет системным администраторам распределять ответственность между пользователями более гибко. *Sudo* контролируется конфигурационным файлом **/etc/sudoers**. Пытаясь произвести впечатление на хорошенькую девушку на вечеринке, я могу сказать что-то вроде «*sudo* предоставляет контролируемое повышение привилегий» (мне никогда не удавалось стать душой вечеринки). Или, говоря проще, он позволяет указать, кому что разрешается делать.

Освоение *sudo* в значительной степени означает освоение синтаксиса файла **/etc/sudoers**. К сожалению, документация для **sudoers** описывает синтаксис файла с использованием расширенной формы Бэкуса-Наура, нотации для формального описания грамматик: если вы не так давно получили ученую степень по компьютерным наукам, все будет отлично, в противном случае – оставьте надежду. Тем не менее, несколько примеров прояснят ситуацию.

В качестве простого примера допустим, что я хочу разрешить пользователю harry изменять системное время и дату из командной строки, используя команду *date*. Команда *date* позволяет любому пользователю, запустившим ее без аргументов, узнать текущее время и дату:

```
$ date
Thu Sep 15 07:56:52 BST 2006
```

А вот если вы попытаетесь изменить время и дату, не будучи root, это вам не удастся:

```
$ date 09150755
date: cannot set date: Operation not permitted
Thu Sep 15 07:55:00 BST 2006
```

Чтобы harry мог изменять дату и время, вы должны сообщить *sudo*, что на запуск *date* у него есть права root. Для этого поместите такую строку в **/etc/sudoers**:

```
harry snowhite=(root) /bin/date
```

В терминологии *sudo* это называется спецификацией привилегий. Давайте рассмотрим каждое поле:

harry Первое поле – учетная запись, которой мы даем особые привилегии.

snowhite Второе поле – имя машины, к которой имеют отношения эти привилегии. Указание конкретного имени позволит вам держать данные **sudoers** на целую когорту машин в одном файле (через NFS, например), и настройка будет у каждой своя. Часто в этом поле можно видеть ALL.

=(root) Эта запись указывает пользователя, чьи права используются для запуска команд.

/bin/date Последнее поле – имя команды, которую может запускать пользователь. Полный путь указывать обязательно.

Для редактирования файла **sudoers** вы должны быть root и использовать при этом команду *visudo* (а не обычный редактор). Наш друг *visudo* заблокирует файл от одновременных обновлений и проверит его синтаксис перед сохранением.

С этой строкой в *sudoers* harry может менять время и дату так:

```
$ sudo date 09150755
```

У harry спросят пароль. Некоторые дистрибутивы Linux включают в **/etc/sudoers** строку вроде этой:

```
Defaults targetpw
```

При наличии этой строки harry должен будет вводить не свой пароль, а пароль того пользователя, от лица которого запускается команда (обычно root). Вы должны удалить эту строку из **sudoers**, чтобы пользователь вводил свой пароль, а не root'а. Иначе будет потеряна вся гибкость ролевой системы. Вы также должны удалить такую строку, если у вас она найдется:

```
ALL ALL=(ALL) ALL
```

Когда harry воспользуется *sudo* и введет пароль, *sudo* на какое-то время позволит ему выполнять команды без проверки пароля. По умолчанию этот промежуток времени равен пяти минутам; вы можете его изменить, добавив в *sudo*ers такую строку:

```
Defaults timestamp_timeout=0
```

Нулевое значение означает, что *sudo* будет спрашивать пароль всегда.

Ролевая игра

Давайте рассмотрим другие особенности **sudoers**. Ключевое слово ALL используется в некоторых местах файла. Например:

```
isaac ALL=(ALL) ALL
```

Эта строка говорит о том, что пользователь isaac может запускать любую программу от лица любого пользователя на любой машине (конечно, если он укажет свой пароль или пароль соответствующего пользователя на нужной машине).

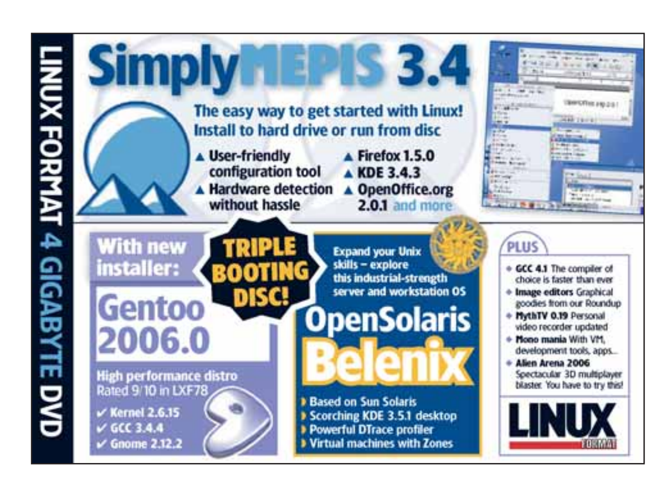
Вы также можете определять псевдонимы (alias). Псевдонимы – это просто именованные списки пользователей, машин или команд. Сами по себе псевдонимы ничего не делают, зато они облегчают написание привилегий для множества пользователей. Покажу вам на примере.

Пусть вы хотите дать определенному пользователю права на управление пользовательскими учетными записями. Для этого есть множество утилит – *useradd*, *groupadd* и т.д. В **sudoers** вы можете определить командный псевдоним (назовем его USERMGMT):

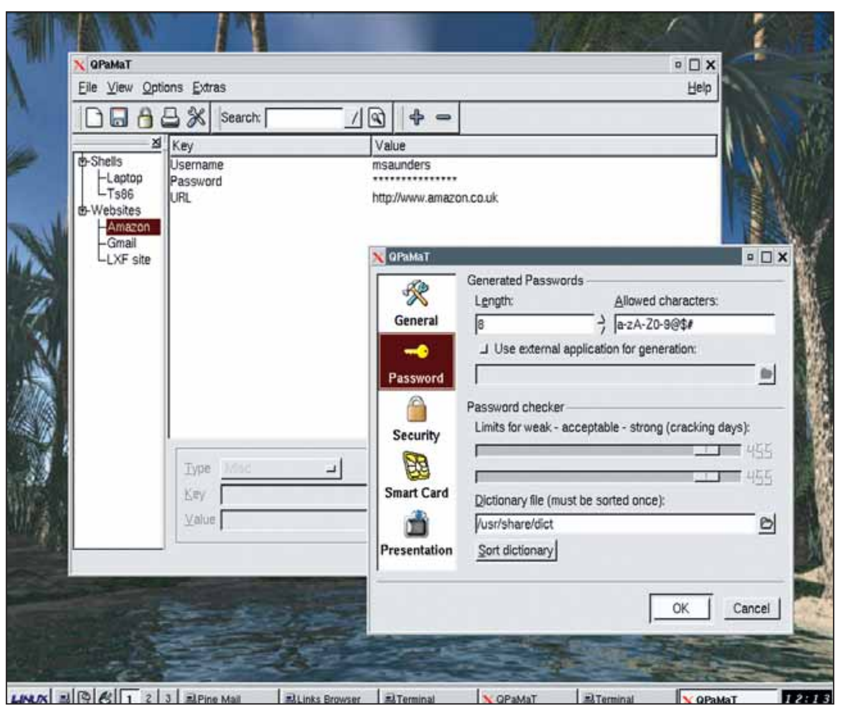
```
Cmnd_Alias USERMGMT=/usr/sbin/useradd,/usr/sbin/usermod,\
    /usr/sbin/groupadd,/usr/sbin/groupmod,\
    /usr/sbin/userdel,/usr/sbin/groupdel
```

Обратите внимание на символ ****, означающий, что строка имеет продолжение. Определив псевдоним, мы можем дать пользователю isaac возможность запускать любую из этих программ таким образом:

```
isaac ALL=(root) USERMGMT
```



› Наши Live-дистрибутивы на вид безобидны, но при физическом доступе к вашей машине кто угодно может получить через них привилегии root.



› *QPaMaT* (живет на http://qpamat.berlios.de) – удобная утилита поддержки паролей на базе *Qt*, позволяющая управлять всеми паролями как бы из одного центра.

Вы можете рассматривать USERMGMT как роль (хотя в документации к *sudo* этот термин не используется), определяющую команды, которые играющий эту роль может выполнить.

Можно определять псевдонимы пользователей (имя перечня пользователей) и присвоить им роли:

```
User_Alias ADMINS tom,joe,sue
ADMINS ALL=(root) USERMGMT
```

Таким способом мы объявили, что tom, joe и sue могут играть роль USERMGMT.

Так же можно определить и список машин, используя диапазон IP-адресов или имена хостов:

```
Host_Alias LOCALNET 192.168.0.0/255.255.255.0
Host_Alias TRUSTEES snowhite,happy,sneezy
```

Теперь мы можем дать определенным пользователям привилегии только на определенных машинах:

```
ADMINS TRUSTEES=(root) USERMGMT
```

Что ж, на сегодня хватит. Вы узнали о том, как работает механизм паролей в Linux, и, надеюсь, до вас дошло, почему необходимо придумывать сильные пароли. Вы увидели, как можно защититься от тривиальной атаки с физическим доступом, установив пароль на *Grub*. Вы узнали, как избежать раздачи пароля root кому попало и использовать *sudo* для того, чтобы позволить определенным пользователям запускать определенные программы с ограниченными привилегиями. LXF



Никогда не выдавайте пароль root. Если кому-то понадобилось ограниченное повышение привилегий, используйте **sudo**.

» Через месяц Мы поотключаем ненужные службы и закрутим гайки прав на работу с файлами.

Ogre: Освещаем

ЧАСТЬ 3: Как превратить травянистый холм в крепость в стиле Тюдоров и создать солнце и луну всего на четырех страницах? У **Пола Хадсона** есть ответ...





Наш эксперт

Пол Хадсон написал три книги по Linux и одну по PHP, он участвует в двух GPL-проектах для Mono на SourceForge. Пол любит Emacs.

В этот урок мы включили много интересного: наша задача – взять пустой ландшафт из прошлого номера, построить дом на плато, добавить света и тени, и, самое интересное, позволить игроку гулять по дому. Легко? Ой, нет. А моя задача – как раз облегчить работу, поэтому приготовьтесь разогнать игру до скорости света!

Мы загружали ландшафт в игру *Висельник Чед* с помощью файла **terrain.cfg**, поставляемого вместе с медиа-пакетом *Ogre*. Всего одной строкой кода создается живописная местность, где можно побродить, но остаются две проблемы: на границах ландшафт обрывается в пустоту и нет места, где можно поставить дом. Поэтому нам надо сначала отредактировать файл ландшафта, пусть запляшет под нашу дудку.

Переместите файл **terrain.cfg** из каталога media в ваш главный каталог, где размещен код. Затем переместите файлы **terrain_detail.jpg**, **terrain.png** и **terrain_texture.jpg** из **media/materials/textures** в ваш главный каталог. Учтите, я сказал «переместите», а не «скопируйте» – не оставляйте оригиналы там, где они находятся, не то они будут читаться вместо ваших собственных файлов. Эти четыре файла определяют, как выглядит и ведет себя ландшафт.

Используемый нами менеджер сцены, ответственный за обработку неба и земли, создает ландшафт по двумерной карте высот (это полутоновой план ландшафта, на котором наивысшие точки показаны белым цветом, а самые низкие – черным). В файле **terrain.cfg** указана карта высот **terrain.png**. Действительный цвет определяется файлом **terrain_texture.jpg**, который «натягивается» на весь ландшафт. Наконец, файл **terrain_detail.jpg** обеспечивает детали, которые будут расположены поверх **terrain_texture.jpg**, чтобы текстура выглядела сложнее, чем она есть на самом деле.

Чего нам не хватает, так это места для нашей постройки. В моей версии **terrain.png** (она находится на диске) я подредактировал карту, чтоб выделить пространство для здания (*см. «Создание карты высот в Gimp», стр. 68*). Вы также заметите, что я сменил цвет **terrain_texture.jpg** на цвет травы, ну устал я от этой пустыни! Теперь откройте в текстовом редакторе файл **terrain.cfg** и найдите строчку #VertexNormal=Yes. Знак # – это комментарий, означающий, что VertexNormal отключен. Удалите #, чтобы включить опцию – это вынудит *Ogre* генерировать нормали для нашего ландшафта, которые понадобятся нам позже (*см. врезку «Зачем нужны нормали» на стр. 67*). В файле **resources.cfg** проверьте, что в секции General указано FileSystem=., это заставит *Ogre* считывать файлы ландшафта из вашего текущего каталога. Вперед! Запустите нашу игру – вы увидите обширную площадку, здесь-то мы и поставим...

Дом, который построил Пол

В *LXF83* я писал, что в игре *Висельник Чед* предусмотрено наличие зданий и возможность игроков входить в них, а там, возможно, вступать в борьбу. Теперь, когда мы разровняли место, пора строить домик в тюдоровском стиле. Чтобы создать сетку дома, нам требуется объект Entity, представляющий наш дом, и SceneNode (оболочка объекта, которая позволит позиционировать дом в пространстве). Мы ставим дом раз навсегда, двигаться ему не надо, поэтому просто объявим Entity в **chad.h**. Добавьте строку в **chad.h** сразу за определением m_OceanFlowNum:

```
Entity* HouseEntity
```



› **Тюдоровский домик днем – дымоход отбрасывает тень на крышу, а внизу слева тень от ландшафта.**

» **Месяц назад** Мы научили нашего героя двигаться и добавили реалистичный эффект воды.



Новый дом

Теперь добавьте этот код в конец метода createScene() файла **chad.cpp**:

```
HouseEntity = m_SceneMgr->createEntity("house", "tudorhouse.
    mesh");
SceneNode* HouseNode = m_SceneMgr->getRootSceneNode()->
    createChildSceneNode("HouseNode");
HouseNode->attachObject(HouseEntity);
HouseNode->translate(1100,137, 500);
HouseNode->scale(0.25,0.125,0.25);
```

Файл **tudorhouse.mesh** – один из стандартных сеточных объектов, входящих в *Ogre*, и он выглядит великолепно. Заметим также, что дом огромный, поэтому я использовал scale() и приспособил его размер к ландшафту. Вот и все, что требовалось для добавления дома в нашу игру, поэтому перезапустите *make* и полюбуйтесь результатом. Нет, определения столкновений пока нет – оставим его на потом!

Небесное вторжение

Сейчас свет в нашей сцене рассеянный, и у сцены скучный и безжизненный вид. Добавлю-ка я немного простых источников, чтобы вы увидели, как работают свет и тени. Упор сделан на слове «простые», потому что *Ogre* плоховато работает с освещением неба и земли.

Потребуется четыре переменных для работы со светом в **chad.h**; добавим их в конец класса CChadGame:

```
Light* m_SunLight;
Light* m_MoonLight;
double m_SunY;
double m_SunZ;
```

Да, мы собираемся создать два света: один для солнца, второй для луны. Для получения простого эффекта смены дня и ночи нам надо вращать два источника света по кругу, чтобы свет плавно перемещался по ландшафту. Также необходимо включить тени. У *Ogre* есть три типа теней и три типа света. Разница между тенями заключается в скорости: тень низкого качества прорисовывается быстрее всего. Первый тип – SHADOWTYPE_TEXTURE_MODULATIVE, использующий простую текстуру для создания теней в нужных местах. На практике тени получаются грубоватые – сгодятся только для тех, у кого слабая видеокарта, все остальные в восторг не придут. Следующий по качеству тип – SHADOWTYPE_STENCIL_MODULATIVE. Тени выглядят более привлекательно, но создаются в один проход, поэтому результат не идеален. Наконец, SHADOWTYPE_STENCIL_ADDITIVE. Тут тени выглядят превосходно, для каждого источника света построение осуществляется за отдельный проход, поэтому пересечение теней от двух источников будет выглядеть темнее, чем остальные части тени. Как вы понимаете, это довольно сложно, но заманчиво!



› **При вызове m_SceneMgr->setShadowDebugShadows(true) *Ogre* покажет, куда предметы отбрасывают тени. Дом отбрасывает тень и от солнца, и от луны.**

Каждый из трех типов источников характеризуется своим способом освещения. Точечный источник излучает свет как Солнце – одинаково во всех направлениях, из определенной точки. Источник направленного света излучает свет в одном направлении и не имеет начальной точки. С нашей позиции на Земле солнце выглядит направленным источником, потому что мы видим его свет, исходящий из одного направления, но не видим его в других направлениях. Наконец, существуют источники узконаправленного света, типа софитов: у них есть начальная точка, направление, их свет слабеет с расстоянием, ширину угла освещения вы назначаете сами.

Оба наших источника, солнце и луна, находятся очень далеко, поэтому они могут считаться источниками направленного света. Не будем пренебрегать деталями – выберем опцию SHADOWTYPE_STENCIL_ADDITIVE, чтобы тени хорошо смотрелись. В последующем коде немного приглушим рассеянное освещение, чтобы заметнее стали дневной и ночной источники света.

Вот код – поместите его в метод createScene():

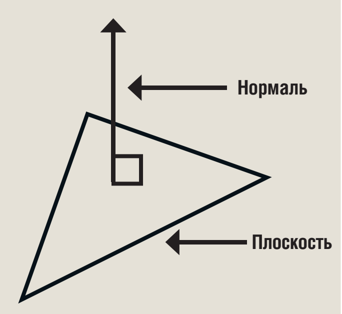
```
m_SceneMgr->setAmbientLight( ColourValue(0.35, 0.35, 0.35) );
    m_SceneMgr->setShadowTechnique(SHADOWTYPE_STENCIL_
    ADDITIVE);
m_SunLight = m_SceneMgr->createLight("Sun");
m_SunLight->setType(Light::LT_DIRECTIONAL);
```

»

Зачем нужны нормали

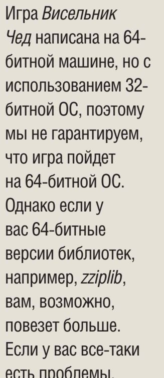
3D-графика составляет рисунки из множества многоугольников, обычно треугольников, потому что они проще. Но как *Ogre* узнает, в каком направлении смотрит треугольник? В нашем понимании, у плоской фигуры две стороны, но используется-то всего одна, поэтому необходимо сообщить Ogre, где у треугольника верх. Благодаря этой информации *Ogre* сможет правильно освещать объекты. Здесь и пригодится нормаль: это вектор, перпендикулярный поверхности и направленный вверх.

При программировании на чистом OpenGL нам пришлось бы бесчисленное множество раз вызывать функцию **glNormal()**, которая определяет нормаль для последующих векторов. Но в *Ogre* достаточно отредактировать **terrain.cfg** и установить VertexNormal на Yes, чтобы *Ogre* сам вычислял нормали к ландшафту. Благодаря этим нормалям наш ландшафт будет правильно освещаться, потому как *Ogre* знает лицевую сторону каждого треугольника.



› **Нормаль сообщает *Ogre*, где у треугольника верх, это позволяет правильно осветить треугольник.**

Скорая помощь



Игра *Висельник Чед* написана на 64-битной машине, но с использованием 32-битной ОС, поэтому мы не гарантируем, что игра пойдет на 64-битной ОС. Однако если у вас 64-битные версии библиотек, например, *zziplib*, вам, возможно, повезет больше. Если у вас все-таки есть проблемы, попробуйте добыть последнюю версию *DevIL* и запустить **ldconfig** от имени root.

```
m_SunLight->setDiffuseColour(1.0, 1.0, 1.0);
m_SunLight->setSpecularColour(1.0, 1.0, 1.0);
m_SunLight->setDirection(Vector3(0, sin(m_SunY), sin(m_
   SunZ)));
m_MoonLight = m_SceneMgr->createLight("Moon");
m_MoonLight->setType(Light::LT_DIRECTIONAL);
m_MoonLight->setDiffuseColour(0.3, 0.3, 0.5);
m_MoonLight->setSpecularColour(0.0, 0.0, 0.0);
m_MoonLight->setDirection(Vector3(0, -sin(m_SunY), sin(m_
   SunZ)));
```

Привести луну и солнце в движение можно, отредактировав метод frameStarted(), чтобы метод setDirection() вызывался для каждого источника. Необходимо также изменить переменные m_SunY и m_SunX, заставив их медленно увеличиваться. Вставьте этот код в начало метода frameStarted():

```
m_SunLight->setDirection(Vector3(0, sin(m_SunY), sin(m_
   SunZ)));
m_MoonLight->setDirection(Vector3(0, -sin(m_SunY), -sin(m_
   SunZ)));
m_SunY += 0.001;
m_SunZ += 0.001;
```

Теперь вы должны увидеть, что здание отбрасывает тень.

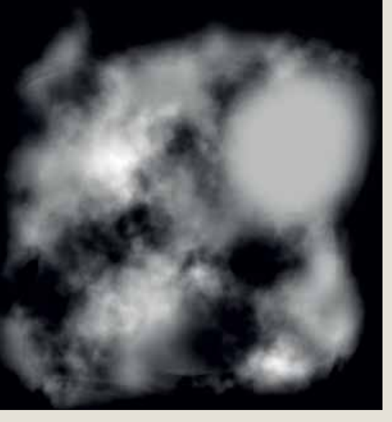
Похождения бравого солдата Квейка

Вы помните, что я собирался строить «Тюдоровскую крепость», но пока что мы имеем только невинного вида домик на плоской траве. На крепость не похоже, правда? Сейчас мы это изменим, соорудив внутри заведомо неприветливую обстановку. На последних трех уроках мы играли в песочнице менеджера сцен *Ogre* для открытого ландшафта, но если наши персонажи намерены разгуливать по зданию, потребуется еще один менеджер сцены – BSP.

Создание карт высот в Gimp

Карты высот – это полутоновые текстуры, которые сообщают Ogre, как надо рисовать ландшафт, где белые точки – это максимальные высоты, а черные – минимальные. Ландшафт, с которым мы работали на последних двух уроках, был уже заготовлен в Ogre, и выглядел он не особо хорошо, поэтому мы использовали *Gimp* и подредактировали края таким образом, чтобы они плавно уходили в море. Используя *Gimp*, я обнаружил, что лучше всего менять высоту с помощью Кисти – вот как она настраивается: в окне Редактор кисти выберите круглую форму и наименьшее значение жесткости, затем измените радиус до нужного вам значения. Установите цвет кисти черным, а затем превратите ваш ландшафт в остров. Для создания площадки я посадил большую светло-серую кляксу в правом верхнем углу карты. Малое значение жесткости позволяет сделать плавный переход от серого к черному, соответствующий на ландшафте пологому холму, куда будет взбираться наш герой.





› Слева представлена исходная карта высот, справа её отредактированный вариант: по краям зачернено, чтоб получился остров, а светлое пятно – площадка для нашего дома.

BSP означает binary space partition [двоичное разбиение пространства] и позволяет очень быстро загружать и обрабатывать закрытые пространства. Карты BSP используется во многих играх, главным образом в тех, что используют движок *Quake*, включая *Jedi Knight*, *Soldier of Fortune*, *Half-life*. Проблема с BSP состоит в том, что он не умеет хорошо обрабатывать открытые пространства (включая наш ландшафт), поэтому текущий менеджер сцены тоже придется оставить.

Добавление нового менеджера сцены вводит интересные осложнения в наш игровой движок. Во-первых, необходимо отредактировать файл **resources.cfg**, чтобы использовался файл карты *Quake 3*. Сотни карт доступны бесплатно через сеть, но на самом деле демо-версии *Ogre* уже поставляются с очень красивой картой *Chiroptera* (доступа для скачивания по адресу http://simland.planetquake.gamespy.com/pages/q3maps/chiroptera.htm), ею мы и воспользуемся.

Во-вторых, необходимо разделить методы createScene() и frameStarted() на CreateIndoorScene(), createOutdoorScene(), frameStartedInside(), frameStartedOutside(). Проще всего это сделать, переименовав createScene() и frameStarted(), например, как createOutdoorScene() и frameStartedOutside(), а затем создав заглушки для createIndoorScene() и frameStartedIndoor(). Также потребуется объявить новые методы в **chad.h**.

Сделав это, добавьте строчку в **chad.h** в конце класса CChadGame:

```
int scenemanager;
```

Переменная будет показывать, снаружи мы или внутри. Теперь можно написать новый метод frameStarted() для автоматического вызова нужного «подметода», нечто вроде такого:

```
bool CChadGame::frameStarted(const FrameEvent& evt) {
switch (scenemanager) {
        case ST_EXTERIOR_CLOSE:
                return frameStartedOutside();
                break;
        case ST_INTERIOR:
                return frameStartedInside();
                break;
}
return true;
}
```

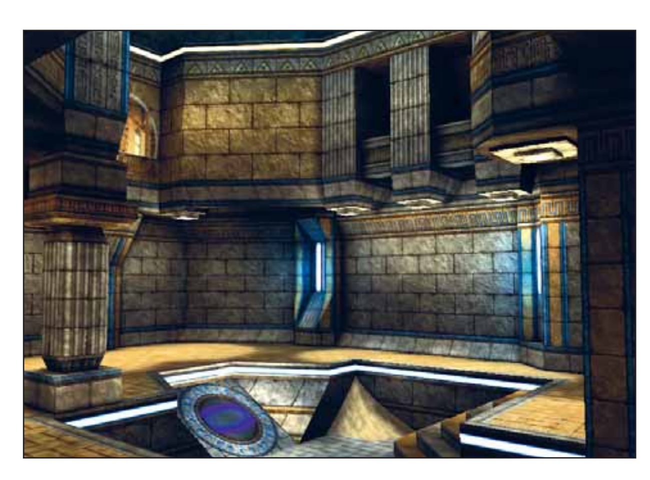
Переменную scenemanager необходимо переустанавливать, как только происходит смена типа сцены. Для открытого пространства необходимо добавить эту строку в начале метода createOutdoorScene():

```
scenemanager = ST_EXTERIOR_CLOSE;
```

Метод frameStartedInside() новый, но ничего особенного не делает – в данный момент он должен представлять урезанную версию метода frameStartedOutside() и обрабатывать простой ввод от клавиатуры. Вот пример подобного кода:

```
bool CChadGame::frameStartedInside() {
m_InputReader->capture();
if(m_InputReader->isKeyDown(KC_ESCAPE)) return false;
Vector3 translateVector = Vector3::ZERO;
float playerspeed = m_Player->getSpeed();
if (m_InputReader->isKeyDown(KC_W)) translateVector.z = -
   playerspeed * 5;
if (m_InputReader->isKeyDown(KC_S)) translateVector.z =
   +playerspeed * 5;
if (m_InputReader->isKeyDown(KC_A)) translateVector.x = -
   playerspeed * 5;
if (m_InputReader->isKeyDown(KC_D)) translateVector.x =
   +playerspeed * 5;
if (m_InputReader->isKeyDown(KC_F1)) createIndoorScene();
if (m_InputReader->isKeyDown(KC_F2)) createOutdoorScene();
m_Camera->moveRelative(translateVector);
return true;
}
```

Вы заметите, что переменная playerspeed умножена на пять – дело в том, что карта *Quake* огромна, и прогулка с прежней скоростью превратилась бы в вечность. Если такое решение вы находите примитив-



› Это не интерьер XVI века, но карта Quake погружает наш дом в такие же потёмки, как во времена Тюдоров.

ным, попробуйте увеличить ландшафт, удалив вызов scale() для дома, а затем выберите значение скорости, пригодное везде.

Теперь перейдем к трудной задаче: загрузке карты *Quake*. Это делается в методе createIndoorScene(), который загрузит нашу карту и создаст новую камеру и область просмотра. Прежде чем перейти к другому менеджеру сцены, необходимо удалить камеры и области просмотра, ассоциированные с текущим менеджером.

Вот этот код:

```
void CChadGame::createIndoorScene() {
scenemanager = ST_INTERIOR;
if (m_SceneMgr != NULL) {
m_Ogre->getAutoCreatedWindow()->removeAllViewports();
            m_SceneMgr->destroyAllCameras();
}
m_SceneMgr = m_Ogre->createSceneManager("BspSceneMan
   ager");
ResourceGroupManager::getSingleton().linkWorldGeometryToR
   esourceGroup(
ResourceGroupManager::getSingleton().getWorldResourceGrou
   pName(), "maps/chiropteradm.bsp", m_SceneMgr);
ResourceGroupManager::getSingleton().initialiseAllResourceGr
   oups();
ResourceGroupManager::getSingleton().clearResourceGroup(Re
   sourceGroupManager::getSingleton().getWorldResourceGroupNa
   me());
ResourceGroupManager::getSingleton().loadResourceGroup(Re
   sourceGroupManager::getSingleton().getWorldResourceGroupNa
   me(), false, true);
m_Camera = m_SceneMgr->createCamera("Player Cam");
m_Camera->setNearClipDistance(4);
m_Camera->setFarClipDistance(4000);
m_Viewport = m_Ogre->getAutoCreatedWindow()
  ->addViewport(m_Camera);
m_Viewport->setBackgroundColour(ColourValue(0, 0, 0));
ViewPoint vp = m_SceneMgr->getSuggestedViewpoint(true);
m_Camera->setPosition(vp.position);
m_Camera->pitch(Degree(90));
m_Camera->rotate(vp.orientation);
m_Camera->setFixedYawAxis(true, Vector3::UNIT_Z);
}
```

Установка значений «ближней» и «дальней» дистанций обрезания необходима, потому что наша камера работает внутри замкнутой области, и нужно фокусироваться на вещах, расположенных близко к камере. Метод getSuggestedViewport() считывает произвольную начальную позицию игрока на карте и устанавливает камеру в эту точку. Нам необходимо установить оси камеры pitch и yaw потому, что *Quake* использует для плоскости пола координаты X и Y, а Z – для высоты, а мы обычно используем для пола X и Z и Y – для высоты.

Мы еще не закончили: понадобится добавить две строки в начало метода createOutdoorScene(), чтобы *Ogre* обновлял все свои ресурсы при загрузке сцены в открытом пространстве. Без них закрытая сцена обновит ресурсы и выгрузит все ресурсы открытой сцены. Итак, добавляем две строки:

```
ResourceGroupManager::getSingleton().clearResourceGroup(Res
   ourceGroupManager::getSingleton().getWorldResourceGroupName());
ResourceGroupManager::getSingleton().initialiseAllResourceGroups();
```

Наконец, проверим, что вызов createScene() в конструкторе CChadGame() (который находится в начале **chad.cpp**) заменен на createOutdoorScene(). Ну вот и все: запустите **make**, затем **chad**. Посмотрим, заметите ли вы разницу....

Сложный вопрос

OK, разницы нет – по крайней мере, с точки зрения игрока. Это потому, что мы написали код обработки для закрытой сцены, но не написали кода, загружающего эту сцену. Вы вольны написать код, способный обнаруживать, что игрок входит в дверь дома, а мы напишем более простую (лишь бы работала) «клавишную реализацию переключения сцен», которую можно использовать в любом месте ландшафта.

У нас уже есть функции обработки клавиатуры frameStaretd*(), поэтому остается добавить в них поддержку двух дополнительных клавиш: F1 (для загрузки закрытой сцены) и F2 (для загрузки открытой сцены). Код, который загружает и выгружает ресурсы, камеры и области просмотра, расположен в методах create*Scene(), поэтому о них беспокоиться не надо. Все, что нам надо сделать, это вызвать требуемый метод create*Scene(),например, так:

```
if (m_InputReader->isKeyDown(KC_F1)) createIndoorScene();
if (m_InputReader->isKeyDown(KC_F2)) createOutdoorScene();
```

Поместите эти две строчки вместе с остальными вызовами isKeyDown(), перезапустите *make*, затем саму игру. На этот раз вы должны попасть на карту *Quake* и исследовать ваш новый интерьер, просто нажав клавишу F1 и пару секунд подождав, пока загрузятся необходимые ресурсы *Ogre*.

Мы рассмотрели в этом руководстве много нового. Труднее всего уместить в голове переход от одного менеджера сцены к нескольким; все остальное сравнительно просто! Я надеюсь, что вам понравилось, как легко *Ogre* создает освещение и тени и работает с уровнями BSP, хотя покамест некоторые недостатки игры могут и раздражать – отсутствие обнаружения столкновений, плохой ввод, небо, на котором не сказывается освещение, и так далее. Часть недостатков вы можете исправить сами, ведь это руководство написано для того, чтобы игру программировали вы, а не я. Но с остальными мы разберемся вместе. До следующего раза! LXF

Благодарности



Графическая карта Nvidia GeForce 7800, используемая для разработки в этой серии учебников, была предоставлена фирмой MSI – спасибо, ребята!

Не моя вина

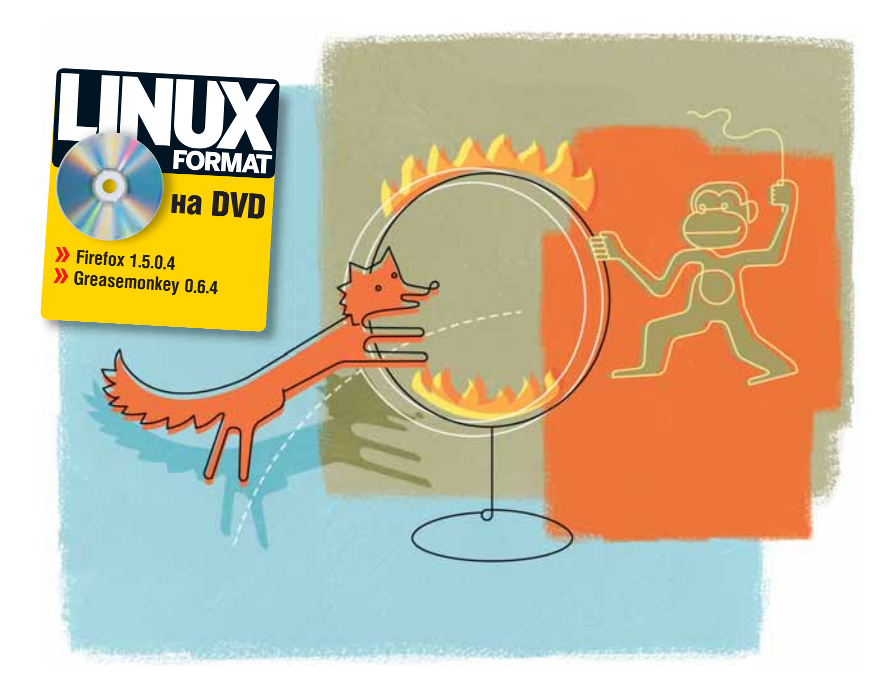
Вы, возможно, заметили довольно уродливые обходные пути в коде этого месяца. Проблема систем типа Ogre, умещающих множество функций всего в паре вызовов методов, состоит в том, что вы толком не знаете, что творится там за кулисами. К сожалению, документация Ogre не идеальна, и иногда приходится действовать наобум.

» **Через месяц** Мы попробуем заняться искусственным интеллектом.



Greasemonkey:

Не чесались ли у вас руки улучшить даже самые отличные web-сайты? Greasemonkey научит сайты работать так, как хочется **Нику Вейчу**, а то и вам.





Наш эксперт

Ник Вейч
Злоупотреблявший интернетом еще с тех пор, как он ходил пешком под стол, наш главный редактор собирается любым способом заставить сайты работать так, как они должны, черт побери!

Задача этого месяца ерундовая: всего-навсего создать собственный web-браузер, способный заново интерпретировать ответившие сайты и заставить их слушаться наших команд.

Думаете, пахать придется серьезно? Ну, мы ж не слабаки! Да и не забывайте: фишка нашей серии в том, что черная работа не для нас, главное, чтоб она была сделана. Так что мы и тут извернемся, как истые хакеры.

Firefox – хороший браузер, более чем хороший. Его нужно лишь чуть-чуть «обработать напильником», например, создать специальное расширение, позволяющее при помощи заготовленного сценария корректировать взятый HTML... Но нам не придется делать даже этого, поскольку – спасибо Аарону Брудману [Aaron Broodman] – на свете есть *Greasemonkey*! Итак, первый этап нашего хитрого плана завершен: мы превратили задачу по написанию миллионов строк кода в задачу состряпать небольшой сценарий JavaScript. Ура!

Хорошо, но прежде чем приступать к созданию сценария, нужно добыть *Firefox* и *Greasemonkey* (и то и другое находится на прилагаемом диске, но проще скачать их прямо из Сети, с сайта http://greasemonkey.mozdev.org). Кстати, в дальнейшем вам не повредит некоторое знакомство с JavaScript, так что взгляните на врезку справа.

Web-приложение, которое я собираюсь улучшать – менеджер фотографий Flickr, а вы можете выбрать любой другой сайт на ваш вкус. Хотя было бы неплохо, если бы этот сайт хорошо поддерживался и имел четкую, понятную иерархическую структуру страниц – например, попытайте счастья с новостным сайтом BBC, с Amazon или с Play. Я выбрал Flickr потому, что здесь есть простор для добавления новых функций, а также потому, что визит на этот сайт – повод оставить парочку глумливых комментариев к чужим фото.

Flickr позволяет пользователям помечать свои фотографии, то есть назначать для них ключевые слова-тэги, по которым затем можно выполнять поиск. Но со страницы каждой конкретной фотографии можно искать только по одному ключевому слову за раз. Если вы крупным планом (макро) засняли цветок наперстянки и выбрали ключевые слова «природа макро наперстянка» ("nature macro foxglove"), то найти похожие кадры, щелкая на метки по отдельности, будет сложновато. Я решил добавить возможность искать по всем имеющимся ключевым словам одновременно. Это хороший тестовый пример, поскольку нам потребуется считывать данные со страницы и записывать их обратно. Начнем с чтения.

Фокус с Flickr

Требуется найти на странице Flickr, где помещено наше фото и информацию о ключевых словах. У JavaScript хватает методов обработки DOM (внутренней иерархии HTML-документа), и все, что нужно сделать – это посмотреть, где в дереве документа находится нужная информация и как однозначно отличить ее от прочего. Проще всего использовать для этих целей встроенный в *Firefox* браузер DOM. Откройте его в Tools – меню инструментов *Firefox* – и пощелкайте по нему, чтобы увидеть дерево иерархии. *Firefox* любезно покажет мигающую красную рамочку вокруг того элемента, который вы выбрали в *DOM Inspector*.

Похоже, что нам повезло. Список ключевых слов на странице фотографии хранятся в элементе <div> и помечен как thetags. Это уникальный идентификатор. Таким образом, мы можем просто перетряхнуть все элементы <div> в документе, пока не найдем нужный. Вот кусочек JavaScript-кода, который это делает:

```
var divs, tagsdiv
divs = document.getElementsByTagName('div');
//Мы сохранили все элементы div как объекты
//в массиве под названием 'divs'
for (var i=0; i<divs.length; i++) {
   if(divs[i].id == 'thetags'){
      tagsdiv=divs[i];
   }
}
```

Здесь использован метод, позволяющий создать массив всех элементов <div>, а затем искать в нем тот, что нам требуется (thetags).

Не знаете JavaScript?

Не беда: если вы хоть раз использовали какой-нибудь язык сценариев, то освоитесь очень быстро. Да и занимаемся мы простейшими вещами – чтением и записью в DOM. Есть много книг и учебников, позволяющих узнать больше, а можно просто взять и разобрать несколько готовых скриптов *Greasemonkey*.

- **JavaScript: The Definitive Guide David Flanagan (O'Reilly)**
- **http://www.htmlgoodies.com/primers/jsp**
- **http://developer.mozilla.org/en/docs/DOM**

» **Месяц назад** Грэм Моррисон использовал PAM-аутентификацию для входа в систему.

Обновим Сеть!

Это не самый эффективный способ, но он годится в качестве примера, и его проще модифицировать для ситуаций, когда требуется найти более одного отмеченного элемента [в противном случае можно было бы использовать код var tagsdiv=document.getElementById('thetags'), – прим.ред.]. Более быстрый метод мы покажем в следующей части.

<div> вообще-то содержит много всего – заголовок, ссылки на иконки, плюс еще по одному <div> на каждый отдельный тэг. И снова мы можем, перебором в цикле, построить список тэгов. Извлечение собственно текста метки достигается выборкой HTML-элемента из нужного узла (какого именно – видно по DOM-структуре). А потом соберем полученные метки в список, который станет основой запроса, который мы передадим обратно на страницу в следующей фазе. Сосредоточьтесь, привожу этот кусок кода:

```
if (tagsdiv) {
  tagsdiv = tagsdiv.getElementsByTagName('div');
  tagslist ='';
  for (var i = 0; i< tagsdiv.length; i++){
    if (i>0) {
      tagslist = tagslist + '+';
    }
    tagslist = tagslist + tagsdiv[i].childNodes[3].innerHTML;
  }
}
```

Метод childNodes любого из узлов дерева содержит список всех дочерних узлов, к которым мы можем обращаться по номерам, чтобы найти нужный и запросить для него значение innerHTML, в котором содержится HTML-представление этого узла. Оно полезно для прояснения непонятных моментов, но проследите за тем, чтобы запросить его для нужного узла, иначе получите описания совершенно посторонних тэгов.

Вернем долги

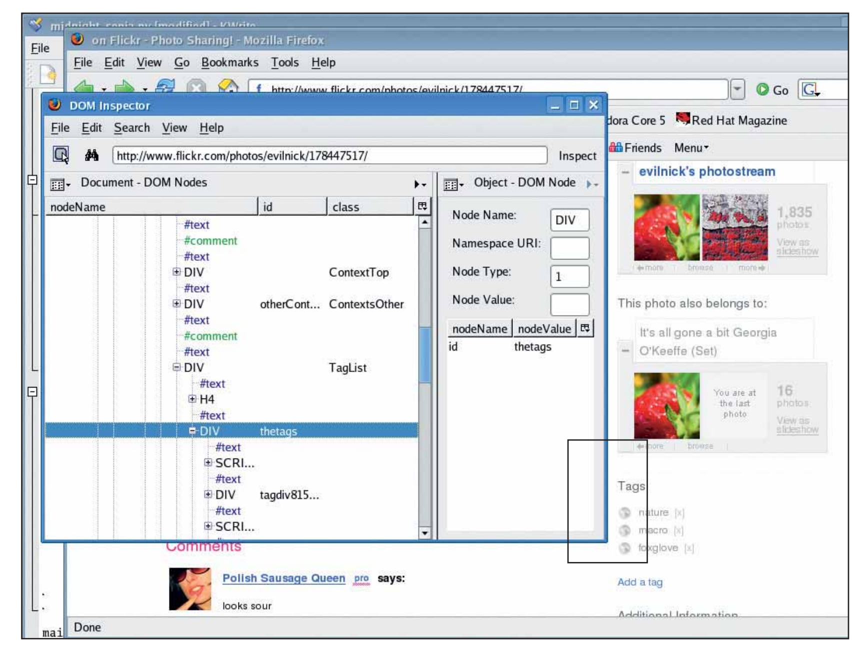
Итак, мы сумели достать нужную информацию из документа. Ура! Настало время поиграть с полученным результатом и записать немножко HTML-текста обратно на страницу. Разработчики JavaScript, видимо, предвидели такую необходимость, и припасли множество встроенных функций. Первый вопрос, который нужно решить – это в какое место страницы поместить наш текст. Подходящим местом выглядит список с информацией о снимке в правой части страницы, ниже списка ключевых слов. Вернувшись в DOM-браузер, мы увидим, что это первый на странице элемент ul. Так что мы просто создаем новый элемент списка и вставляем его в нужное место:

```
var searchtagLi = document.createElement("li");
searchtagLi.setAttribute("class","Stats");
searchtagText = 'Search <a href="http://www.flickr.com/search/
?w=all&q='+tagslist+'"style="text-decoration: none;">similar tags</a> on
Flickr';
searchtagLi.innerHTML = searchtagText;
insertpoint = document.getElementsByTagName("ul")[0];
insertpoint.appendChild(searchtagLi);
```

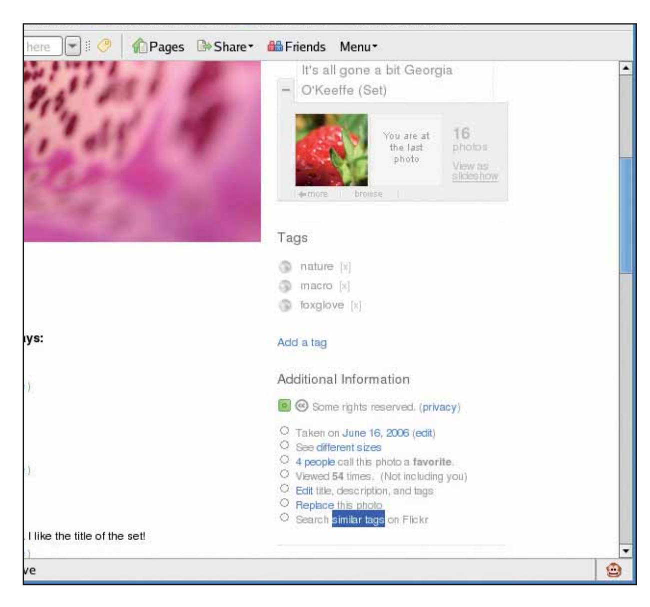
Метод setAttribute используется для указания, какого класса должен быть наш новый элемент, чтобы по CSS-стилю не отличаться от своих соседей.

Текст, который мы добавили – всего лишь ссылка на функцию продвинутого поиска Flickr, которой мы, вместо одного ключевого слова, передаем собранный нами список. Легко видеть, что потом использован метод innerHTML для записи данных в объект. Объект insertpoint указывает на ранее найденный элемент ul, так что остается только вызвать метод appendChild для добавления его в дерево (причем в самый конец списка, что очень правильно).

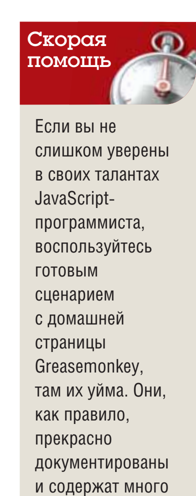
Вот и все, и этого достаточно. Чтобы загрузить скрипт *Greasemonkey*, просто активируйте это расширение и укажите браузеру на файл (назвав его как-нибудь вроде **something.user.js**) – *Greasemonkey* обнаружит файл и позволит вам его загрузить. LXF



› **DOM Inspector великолепен: не только отображает дерево документа (где скрывается моя наперстянка), но и подсвечивает соответствующие области на странице.**



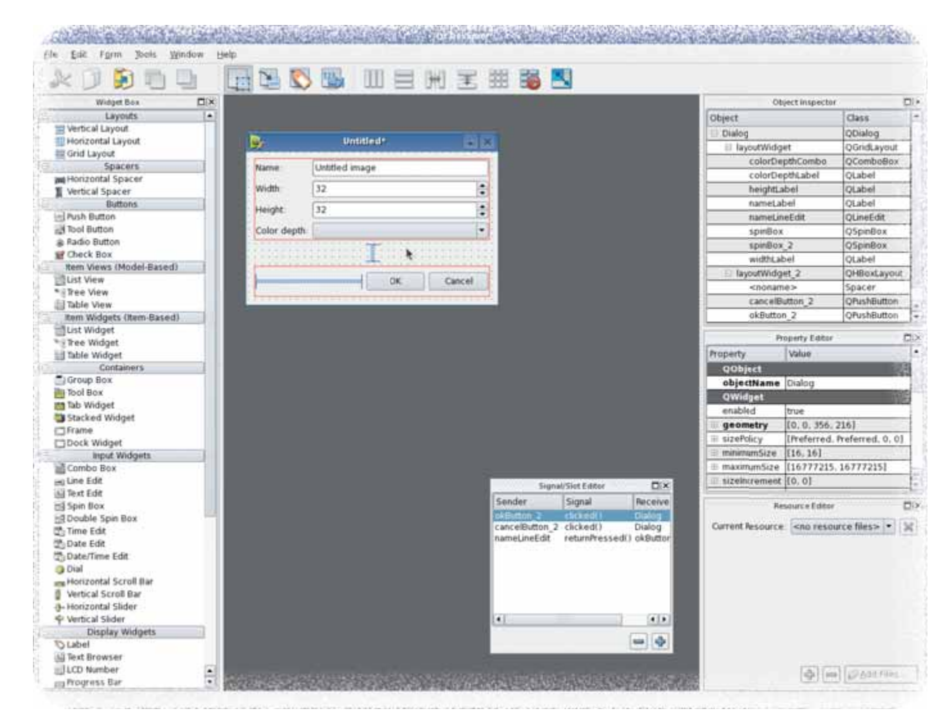
› **Миссия закончена. Небольшая хитрость – и наша добавка (выделена синим) выглядит так, словно и всегда была здесь.**



Если вы не слишком уверены в своих талантах JavaScript-программиста, воспользуйтесь готовым сценарием с домашней страницы Greasemonkey, там их уйма. Они, как правило, прекрасно документированы и содержат много ценных находок.

Модули KDE

ЧАСТЬ 7: Модули, расширения, расширения расширений.... всKIPIте от напряжения и нетерпения под присмотром **Андрея Боровского**!



Модули KDE

Модули KDE (KDE plugins) представляют собой логическое развитие идеи объектно-ориентированной оболочки. Читатели этой статьи наверняка знакомы с базовыми концепциями модулей как средств расширения уже существующих приложений. Механизм модулей позволяет нам не умножать сущностей (приложений) без необходимости всякий раз, когда нам требуется новая функциональность. По самой своей природе модули тесно связаны с теми приложениями, чьи возможности они расширяют. Нельзя написать «просто модуль». Мы рассмотрим программирование модулей для наиболее популярных приложений KDE и начнем с написания модуля для компонента *KParts*.

К-Части

Модули *KParts* широко применяются, например, в браузере *Konqueror*. Сами компоненты *KParts* представляют собой еще один способ расширения возможностей приложения. Примером их использования являются встроенные в *Konqueror* утилиты просмотра файлов различных типов. Каждый раз, когда *Konqueror* открывает какой-либо файл для просмотра, он загружает соответствующий компонент, который встраивается в интерфейс браузера и отображает содержимое файла. Благодаря динамическим компонентам *KParts, Konqueror* выполняет функцию универсального просмотрщика файлов, сохраняя при этом простой дизайн и небольшие размеры основного приложения. Следует отметить, что универсализм *Konqueror* иногда подвергается критике, поскольку в Unix традиционно преобладает другой подход – использование большого числа независимых приложений, каждое из которых делает что-то одно. Мы оставим в стороне вопрос о том, нужно ли писать модули, и ответим на вопрос, как это можно сделать. Модули *KParts* расширяют возможности компонентов *KParts*, то есть, с точки зрения основной программы *Konqueror*, модули являются расширениями расширений. Поддержка модулей *KParts* возможна, естественно, не только в *Konqueror*, но и в любом приложении, поддерживающем соответствующие компоненты *KParts*.

Возможно, на этом этапе у читателя возникает вопрос – а зачем вообще нужны модули *Konqueror*? Необходимость в механизме модулей была бы очевидна, если бы *Konqueror* был закрытым приложением, однако, при наличии исходных текстов, мы можем вносить изменения непосредственно в код компонентов *KParts*. Это верно, и тем не менее модули оказываются полезны и в случае открытого приложения. Главное преимущество использования модулей заключается в лучшей структурной организации кода. Нам легче писать и отлаживать модули, связанные с приложением с помощью строго определенного интерфейса, нежели вносить изменения в исходные тексты приложения, написанного другими программистами. К этому следует добавить и чисто организационный момент – для того, чтобы изменения, сделанные вами в приложении, попали в его основную версию, они должны быть одобрены другими разработчиками, тогда как в случае с модулями у вас есть возможность самостоятельно решить, какие функции являются полезными, а какие – нет.

Что же представляет собой модуль *KParts* с точки зрения взаимодействия с приложением? Модуль – это динамическая библиотека (файл ***.so**), которая загружается приложением, использующим компонент *KParts*. Модуль может добавлять собственные элементы управления в меню и на панель инструментов приложения. В некоторых случаях модули также изменяют содержимое главного окна приложения (которое в этот момент управляется соответствующим компонентом *KParts*).

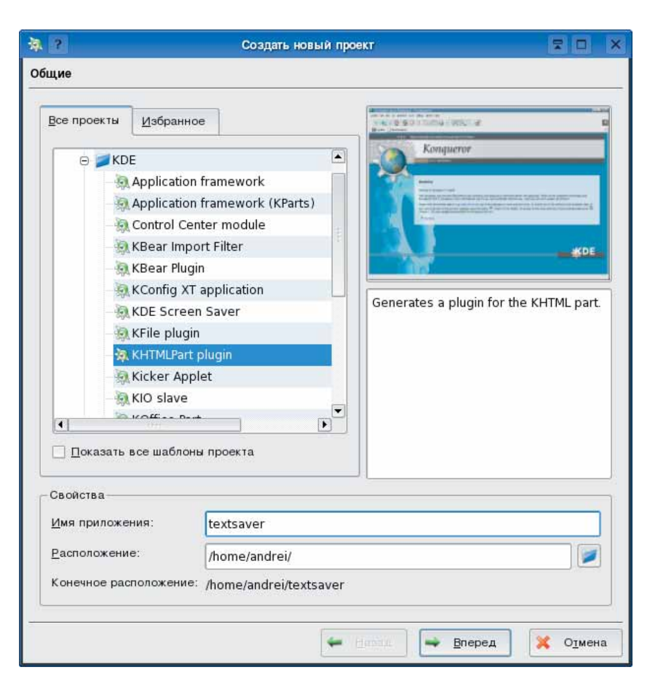
В качестве примера мы напишем модуль для компонента KHTMLPart. Как нетрудно догадаться, этот компонент используется для просмотра страниц HTML. Фактически KHTMLPart представляет собой полноценный встраиваемый HTML-браузер с поддержкой JavaScript. Класс KHTMLPart является потомком класса KParts::ReadOnlyPart, то есть он предназначен для просмотра, а не для модификации открываемого файла. Наш модуль будет сохранять открытую в компоненте KHTMLPart HTML-страницу в простом текстовом формате.

Модуль для KHTMLPart удобен в качестве примера потому, что в *KDevelop* есть шаблон проекта такого модуля. В окне Создать новый проект **(Рис. 1)** мы выбираем пункт KHTMLPart Plugin. Назовем наш проект «textsaver» (полные исходные тексты модуля вы найдете на диске, в файле **textsaver.tar.gz**). Наш модуль, как таковой, состоит только из двух классов – Plugintextsaver (основной класс модуля) и textsaverFactory (вспомогательный класс). Оба класса реализованы в файлах **plugin_textsaver.h** и **plugin_textsaver.cpp**. Прежде чем вносить изменения в классы, добавим в файл **plugin_textsaver.cpp** дополнительные заголовочные файлы:

```
include <stdio.h>
#include <kfiledialog.h>
#include <qcstring.h>
```

Месяц назад Мы завершили наше первое настоящее KDE-приложение.





› (Рис. 1) Создаем проект модуля KParts.

Рассмотрим сначала класс textsaverFactory, являющийся потомком класса KParts::Factory и управляющий созданием объекта класса Plugintextsaver (т.е., этот класс реализует фабрику объектов). Такой объект создается методом textsaverFactory::createObject(), который сгенерирован автоматически. Нам вообще ничего не нужно было бы менять в классе textsaverFactory, если бы не чистый виртуальный метод createPartObject(). Сам по себе он нам не нужен, но для того, чтобы наш модуль мог скомпилироваться, мы должны создать его реализацию. Странно, что *KDevelop*, по крайней мере в версии 3.4.2, не делает этого автоматически:

```
KParts::Part* createPartObject(
  QWidget*, const char*, QObject*,
  const char*, const char*, const QStringList&)
{return 0;}
```

Помимо прочего, разделяемая библиотека, содержащая наш модуль, должна экспортировать функцию, создающую объект класса textsaverFactory. Эта функция, сгенерированная автоматически, экспортируется в формате C:

```
extern "C"
{
 void* init_libtextsaverplugin()
 {
   KGlobal::locale()->insertCatalogue("textsaver");
   return new textsaverFactory;
 }
}
```

Приложение, вызывающее функцию init_libtextsaverplugin(), знает, что делать с возвращенным указателем.

Перейдем к классу Plugintextsaver. Приложение, а точнее, компонент *KHTMLPart*, взаимодействует с нашим модулем при помощи вызова трех методов этого класса – конструктора, деструктора и метода slotAction(). Метод slotAction() представляет собой слот, который вызывается всякий раз, когда пользователь выбирает пункт меню или нажимает кнопку, связанную с модулем. Этот метод содержит код, который, собственно, и отвечает за выполнение команды. В конструкторе класса Plugintextsaver создается объект-действие:

```
Plugintextsaver::Plugintextsaver( QObject* parent, const char* name )
  : Plugin( parent, name )
{
   (void) new KAction( i18n("Save as &Text"), "textsaver", 0,
                  this, SLOT(slotAction()),
                  actionCollection(), "plugin_action" );
}
```

Этот объект будет добавлен в коллекцию действий приложения, вызвавшего компонент KHTMLPart, загрузивший наш модуль. Тем, кто не совсем понимает, о чем идет речь, рекомендуем прочитать позапрошлую статью этого цикла (*LXF82*), в которой мы познакомились с объектами-действиями. Мы указываем объекту-действию имя соответствующего элемента управления и слот, который должен обрабатывать команду. Очевидно, что в общем случае мы можем зарегистрировать несколько действий для нашего модуля. Нам осталось написать только сам метод slotAction():

```
void Plugintextsaver::slotAction()
{
  if ( !parent()->inherits("KHTMLPart") )
  {
     QString title( i18n("HTML Only"));
     QString text(i18n("Only HTML documents can be saved with this plugin."));
     KMessageBox::sorry( 0, text, title );
     return;
  }
  KHTMLPart *part = dynamic_cast<KHTMLPart *>(parent());
  QString FN = KFileDialog::getSaveFileName("", "", 0, "Save Text File As");
  if (FN == "") return;
  QString Cmd = "html2text > " + FN;
  QCString Text = part->documentSource().local8Bit();
  FILE * F = popen(Cmd.ascii(), "w");
  fwrite((char*) Text.data(), 1, Text.length(), F);
  pclose(F);
}
```

У модуля должна быть какая-то возможность взаимодействовать с другими элементами приложения-хозяина (этот термин не следует понимать в том смысле, что наш модуль паразитирует на браузере *Konqueror*). Такая возможность у модуля есть, благодаря тому, что конструктору объекта модуля был передан указатель на вызвавший его объект класса KTHMLPart. Доступ к этому указателю можно получить с помощью метода parent(). Первый блок кода в методе slotAction() проверяет, действительно ли объект parent() принадлежит классу KHTMLPart. На практике это излишне, поскольку модуль никогда не будет вызван для какого-либо иного компонента. Далее мы получаем указатель на родительский объект KHTMLPart, запрашиваем у пользователя имя файла для сохранения текста (надеюсь, вы еще не забыли, что должен делать наш модуль) и, с помощью метода documentSource() получаем HTML-текст страницы, открытой компонентом KHTMLPart. Метод documentSource() возвращает данные в строке QString в двухбайтовой кодировке. Мы преобразуем полученную строку в локальную однобайтовую кодировку и сохраняем результат в переменной **Text** типа QCString. Далее, с помощью функции **popen(3)**, мы запускаем внешнюю утилиту *html2text*, преобразующую данные HTML в простой текст. Этой утилите мы передаем содержимое переменной **Text** и имя файла для сохранения результата. Применяемый нами прием – использование в программе внешней утилиты для решения нашей задачи, очень широко распространен в Unix-системах.

Хотя мы и определили все методы классов нашего модуля, работа над ним еще не закончена. Модуль создает новый объект-действие для вызвавшего его приложения, но приложение пока не знает, как оно должно (и должно ли) отображать соответствующий этому действию элемент интерфейса. Наш модуль может дополнить пользовательский интерфейс хозяйского приложения с помощью механизма XMLGUI, описанного в прошлой статье (*LXF83*). Откройте файл **plugin_textsaver.rc**:

```
<!DOCTYPE kpartgui>
<kpartplugin name="textsaver" library="libtextsaverplugin" version="1">
<MenuBar>
 <Menu name="tools"><Text>&amp;Tools</Text>
  <Action name="plugin_action"/>
 </Menu>
</MenuBar>
```

»

```
<ToolBar name="extraToolBar">
  <Action name="plugin_action"/>
</ToolBar>
</kpartplugin>
```

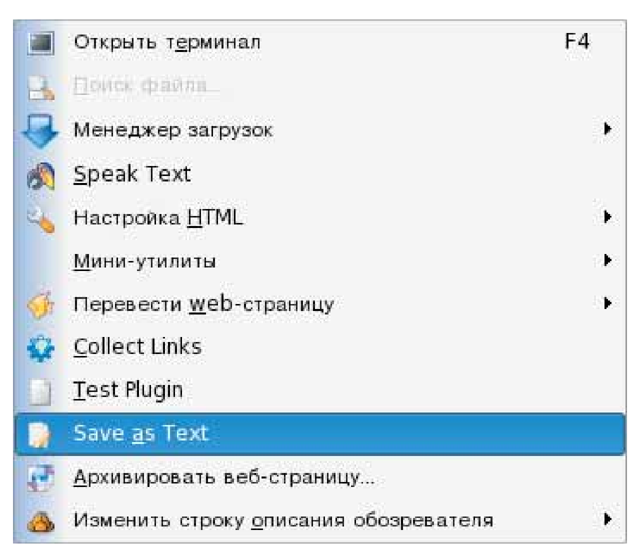
Те, кто читал предыдущую статью, не найдут в этом тексте ничего неожиданного. Мы регистрируем два элемента управления – для меню и панели быстрого доступа. Команда вызова нашего модуля (Save as Text) будет расположена в меню Tools (Сервис), хотя, возможно, следовало расположить ее в меню Файл. Кнопка команды по умолчанию будет расположена на дополнительной панели инструментов, которая (опять-таки, по умолчанию) не видна.

Нам осталось скомпилировать и установить наш модуль. Обычно модули устанавливаются в общую системную директорию, например, в **$KDEDIR/lib/kde3/**, так что для установки понадобятся права root. При отладке модуля следует учитывать, что текущая версия модуля кэшируется оболочкой *Konqueror* (и даже *ldconfig* не помогает), так что для того, чтобы увидеть изменения, внесенные после повторной установки расширения, вам, возможно, придется перезапускать оболочку.

Скомпилируйте и установите модуль, после чего запустите новый экземпляр Web-браузера (но не файл-менеджера) *Konqueror* или откройте HTML-страницу в файл-менеджере. В меню Сервис вы увидите команду, вызывающую наш модуль **(Рис. 2)**. Выше уже было сказано, что система следит за тем, чтобы наш модуль не был вызван для неподходящего компонента. Это выражается в том, что соответствующая команда меню видна тогда, когда мы открываем HTML-страницу, и не видна в остальных случаях.

Если при просмотре сохраненной нашим модулем страницы в каком-нибудь текстовом редакторе вы увидите нечто странное, то не спешите ругаться. Используемая нами утилита *html2text,* во-первых, пытается сохранить расположение текста, соответствующее его расположению на странице HTML, а во-вторых, сохраняет текст в формате команды cat (в этом формате гиперссылка выделяется повторением каждой буквы ее надписи по два раза).

Помимо модулей *KParts,* браузер *Konqueror* поддерживает еще несколько типов модулей. Модули панели навигации позволяют добавить новые команды на панель навигации *Konqueror* (эта панель по умолчанию припаркована к левому краю окна менеджера файлов). Все, что требуется от модуля панели навигации – возвратить указатель на визуальный элемент, который и будут отображен браузером при вызове соответствующей команды. Модули *KFile* позволяют добавлять элементы в окно свойств файла (это окно можно вызвать с помощью команды Свойства контекстного меню менеджера файлов). Обычно модули *KFile* используются для вывода метаданных файлов. Поскольку для всех мыслимых типов файлов такие модули уже написаны, вам не придется писать модуль *KFile*, если только вы не разрабатываете собственный файловый формат.



› (Рис. 2) Команд вызова модуля.

Модули KIPI

Структура модуля обычно настолько тесно связана со структурой приложения, для которого этот модуль предназначен, что изучение программирования модулей имеет смысл только в контексте изучения разработки для конкретного приложения. Однако из этого правила есть и исключения. Примером модулей, которые могут использоваться для работы с несколькими разными приложениями, являются модули *KIPI*. KIPI (KDE Image Plugin Interface – интерфейс KDE для обработки изображений) – это проект, целью которого является создание единого интерфейса модулей для основанных на KDE приложений, предназначенных для просмотра и редактирования растровых изображений, – *Digikam*, *KimDaBa* (ныне *KPhotoAlbum*), *ShowFoto* и *Gwenview*. Хотя «KIPI» представляет собой аббревиатуру, на сайте **extragear.kde.org/apps/kipi/** используется написание «Kipi». Мы будем писать KIPI, дабы читатели понимали, что это аббревиатура, а не русский глагол, написанный транслитом. Модули, соответствующие требованиям KIPI, должны работать одинаково во всех перечисленных приложениях. Думаю, не нужно распространяться о том, насколько это упрощает жизнь программиста, стремящегося к тому, чтобы его разработками могли пользоваться поклонники всех графических программ KDE. Конечно, не все модули могут иметь смысл в контексте каждого приложения. Это становится очевидно, если учесть, что в то время как одни приложения KDE (например, *Gwenview*) обладают исключительно функциями просмотра, то другие (*Digikam*) включают еще и базовые функции редактирования. Если же все программы будут эволюционировать в одном и том же направлении, система утратит разнообразие и само наличие нескольких независимых программ потеряет смысл.

В отличие от других интерфейсов KDE, интерфейс KIPI сравнительно слабо документирован. Это значит, что наилучшим источником сведений о программировании модулей KIPI являются исходные тексты модулей, написанных другими программистами.

Информацию о модулях KIPI можно получить на сайте **extragear.kde.org**. Там же можно загрузить и исходные тексты модулей (уже модифицированные тексты вы можете найти на прилагаемом диске, в файле **kipi-plugins-0.1.2m.tar.gz**). Прежде чем устанавливать модули необходимо установить библиотеку *libkipi* (ее исходные тексты тоже есть на диске, в файле **libkipi-0.1.4.tar.bz2**).

Рассмотрим структуру модуля KIPI на примере модуля *HelloWorld*, который, как следует из названия, как раз и должен служить примером для начинающих модулеписателей (исходные тексты этого модуля находятся в директории **kipi-plugins-0.1.2/kipi-plugins/helloworld/** файла **kipi-plugins-0.1.2m.tar.gz**). Как вы, наверное, догадались, модуль KIPI представляет собой разделяемую библиотеку. Явным образом эта библиотека экспортирует один-единственный класс – класс модуля. Определение соответствующего класса Plugin_HelloWorld мы найдем в файлах **plugin_helloworld.cpp** и **plugin_helloworld.h**.

Пробуем разобраться в том, что делает каждый метод этого класса. Конструктор Plugin_HelloWorld вызывает конструктор базового класса всех классов модулей KIPI KIPI::Plugin и выводит сообщение о том, что модуль загружен. Отметим здесь, что поскольку модули подключаются динамически, вывод такого сообщения полезен не только в процессе отладки модуля. Рассмотрим метод setup(), который, как можно догадаться из названия, будет вызван приложением-хозяином модуля сразу после создания объекта.:

```
void Plugin_HelloWorld::setup( QWidget* widget )
{
   KIPI::Plugin::setup( widget );

   // this is our action shown in the menubar/toolbar of the mainwindow
   m_actionHelloWorld = new KAction (i18n("Hello World..."),
                          "misc",
                          0,        // do never set shortcuts from plugins.
                          this,
                          SLOT(slotActivate()),
```

```
                    actionCollection(),
                    "helloworld");

  addAction( m_actionHelloWorld );

  m_interface = dynamic_cast< KIPI::Interface* >( parent() );

  if ( !m_interface )
       {
       kdError( 51000 ) << "Kipi interface is null!" << endl;
       return;
       }
}
```

Этот метод вызывает одноименный метод базового класса и создает объект-действие (таких объектов может быть зарегистрировано несколько). В методе setup() модуль получает также указатель на объект класса KIPI::Interface, который реализует интерфейс взаимодействия между модулем и программой-хозяином.

Далее в классе Plugin_HelloWorld определен метод slotActivate(), который и выполняет всю работу модуля (именно этот слот был указан при создании объекта-действия). Метод slotActivate() мы пока пропустим и рассмотрим последний метод класса Plugin_HelloWorld::category():

```
KIPI::Category Plugin_HelloWorld::category( KAction* action ) const
{
  if ( action == m_actionHelloWorld )
    return KIPI::IMAGESPLUGIN;

  kdWarning( 51000 ) << "Unrecognized action for plugin category
identification" << endl;
  return KIPI::IMAGESPLUGIN; // no warning from compiler, please
}
```

С помощью этого метода программа-хозяин может выяснить, к какой категории должен принадлежать элемент управления, который она создает для данного объекта-действия. Модуль *HelloWorld* возвращает значение KIPI::IMAGESPLUGIN, которое указывает, что соответствующий элемент управления должен относиться к категории Images.

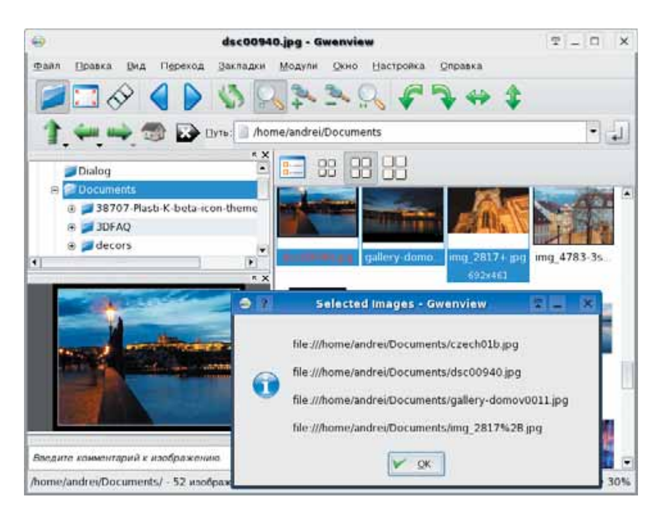
Для того чтобы написать свой модуль KIPI, нам, фактически, необходимо только переписать метод slotActivate() модуля *HelloWorld*. В архиве **kipi-plugins-0.1.2m.tar.gz** вы найдете модуль *mykipi*, который и был создан на основе модуля *HelloWorld* (поскольку заготовки проекта *KDevelop* для такого модуля в KDevelop нет, его проще распространять как часть пакета *kipi-plugins*). Все, что отличает модуль *mykipi* от модуля-прародителя, сосредоточено в методе slotActivate() класса Plugin_MyKIPI:

```
void Plugin_MyKIPI::slotActivate()
{
  kdDebug( 51000 ) << "Plugin_MyKIPI slot activated" << endl;

  KIPI::ImageCollection selection = m_interface->currentSelection();

  if ( !selection.isValid() ) {
     kdDebug( 51000) << "No Selection!" << endl;
  }
  else {
     KURL::List images = selection.images();
     QStringList SL = images.toStringList();
     QString S = SL.join("\n");
     KMessageBox::information(0, S, "Selected Images");
  }
}
```

Метод начинает работу с вывода диагностического сообщения. Затем мы создаем объект класса KIPI::ImageCollection, который содержит ссылки на объекты, представляющие изображения, выбранные пользователем в окне графической программы (ваш разум еще не кипит от этих модулей KIPI?). Если метод currentSelection() вернул значение NULL, значит, пользователь не выбрал ни одного изображения и тогда нашему модулю просто нечего делать. Впрочем, вызов модуля при отсутствии выбранных изображений невозможен, как невозможен вызов модуля *KParts* для неподходящего компонента. Как и положено всякому учебному примеру, модуль mykipi не делает ничего полезного. С помощью метода selection.images() мы получаем список URL выделенных изображений (не забывайте, что мы имеем дело с программами, предназначенными, в основном, для просмотра и редактирования существующих графических файлов). Далее мы преобразуем список URL в QStringList и затем QString, и выводим его содержимое в диалоговом окне.



(Рис. 3) Модуль mykipi в работе.

Когда я писал, что у нашего модуля есть только один класс, я немного упростил ситуацию. На самом деле, модуль реализует еще один класс, а именно класс Factory, управляющий созданием основного объекта класса модуля, однако генерация этого класса выполняется автоматически с помощью определения типа и вызова макроса:

```
typedef KGenericFactory<Plugin_MyKIPI> Factory;
K_EXPORT_COMPONENT_FACTORY( kipiplugin_mykipi,
                    Factory("kipiplugin_mykipi"));
```

Прежде, чем мы установим наш модуль, нам следует отредактировать файл **kipiplugin_mykipi.desktop**. Ниже приводится его отредактированный текст (структура файлов ***.desktop** похожа на структуру файлов ***.ini** Windows, и это не случайное совпадение):

```
[Desktop Entry]
Encoding=UTF-8
Name=MyKIPI
Name[cs]=My KIPI
Comment=Sample KIPI Plugin
Type=Service
ServiceTypes=KIPI/Plugin
X-KDE-Library=kipiplugin_mykipi
author=Andrei Borovsky, borovsky@tochka.ru
```

Файл **kipiplugin_mykipi.desktop** содержит сведения о типе модуля (ServiceTypes=KIPI/Plugin), а также имя модуля и комментарий.

Теперь мы можем скомпилировать и установить модуль (естественный путь – **make install**). Запустим программу *Gwenview* и в меню Модули, в группе Изображения (Images) мы увидим команду Sample KIPI Plugin, которая будет работать так, как мы и ожидаем **(Рис. 3)**.

В этой статье мы рассмотрели, среди прочих, модули для компонентов *KParts*. В следующей, заключительной части серии мы узнаем, как создавать собственные компоненты (и модули для них).

Через месяц Мы научимся создавать собственные компоненты.

Python: Обработка

ЧАСТЬ 4: Язык Python может многое, хотя, конечно, не всё. Но благодаря «привязкам» к таким мощным библиотекам, как GTK и модулям сторонних разработчиков его возможности становятся практически безграничными, пишет Сергей Супрунов.



Рассказывать о различных «расширениях» *Python* можно бесконечно. Чтобы все-таки завершить серию и дать вам возможность погрузиться в самостоятельное изучение этого языка, я решил коротко остановиться на двух сторонних библиотеках, значительно расширяющих возможности *Python* по работе с изображениями и звуком.

Где же кружка?..

Начнем с PIL – *Python* Image Library. Вам наверняка удастся без проблем инсталлировать ее с помощью менеджера пакетов вашего дистрибутива. Не исключено, что она уже установлена (как это имеет место быть в Ubuntu). Но если вам не очень повезло (мало ли, вдруг вы – фанат Slackware), то забрать архив с ее исходным кодом можно на www.Pythonware.com.

Итак, зачем она нужна? Представьте, что вы привезли с моря четыре «флешки», забитых замечательными фотографиями, и хотели бы згрузить их на свою домашнюю страничку... Начинание благое, но каково будет ее посетителям «тянуть» добродушно созданные вашей «Минолтой» файлы по 5 МБ каждый? Принципы гуманизма требуют предварительно «сжать» изображения до приемлемых размеров. Да и миниатюры создать было бы неплохо, чтобы можно было окинуть взглядом сразу несколько фотографий.

Если вы уже собрались запускать Gimp – одумайтесь! Ну на 10 изображений у вас терпения хватит, ну на 100... А если их 1000? Нет, ручная работа не для нас. Вот тут-то и пригодится PIL, где уже реализованы широчайшие возможности по обработке изображений.

Небольшой практикум

Итак, после установки библиотеки все ее модули вы, скорее всего, найдете в **/usr/lib/Python2.4/site-packages/PIL**. Кстати, обычно при инсталляции PIL добавляет в список sys.path также и указанный путь, так что для подключения, скажем, модуля *ImageDraw*, можно будет использовать не только *import PIL.ImageDraw*, но и просто import *ImageDraw*.

Сами модули не слишком щедро прокомментированы, но зато по адресу http://www.Pythonware.com/library/pil/handbook вы найдете превосходное руководство.

Несмотря на множество модулей, входящих в состав библиотеки, для большинства задач вполне хватает модуля Image, а он уже сам неявно задействует возможности остальных. Основой является объект-изображение, которое можно загрузить из файла (используя метод open()), либо создать с нуля (конструктор new()), причем при загрузке файла библиотека сама позаботится о распознавании формата изображения.

Для примера напишем небольшой конвертер, который мог бы брать указанные с помощью шаблона файлы и сохранять их измененные копии в другом каталоге. PIL предоставляет много методов для модификации изображений, в нашем примере мы остановимся на двух – изменении размера и формата. Заодно вспомним, как работать с файловой системой, и познакомимся с небольшим, но полезным модулем getopt для разбора параметров, передаваемых сценарию из командной строки. Код сценария представлен в листинге **conv2.py**.

В строках 8–13 мы используем функцию getopt() одноименного модуля для того, чтобы получить передаваемые в скрипт опции. Первым параметром функция получает список аргументов командной строки (без первого элемента, соответствующего имени самого скрипта), а вторым – список так называемых «коротких» опций, которые нужно найти. (Модуль поддерживает также и «длинные» опции в формате GNU – смотрите документацию или код самого модуля). Если опция задается как логическая (т.е. важно лишь то, присутствует ли она в командной строке), то в списке просто указывается соответствующая ей буква. Если опция параметрическая (т.е. должна сопровождаться каким-то значением), после ее буквы ставится двоеточие. В нашем примере все опции параметрические: -t (формат выходных файлов), -o (каталог для их размещения), -h (высота модифицированного изображения), -w (ширина изображения).

Возвращает getopt() два списка: список распознанных опций в формате [(опция, значение), (опция, значение), (. . .)] (значение указывается только для параметрический опций, для логических второй элемент кортежа остается пустым), и список оставшихся (не распоз-

Месяц назад Мы подключались из Python к базам данных.



графики и звука

Листинг conv2.py

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-

3  import os, sys, getopt, glob
4  import PIL.Image as pim

5  type = "JPEG"
6  outdir = "."
7  width = height = 0

8  opts, pattern = getopt.getopt(sys.argv[1:], "t:o:w:h:")
9  for opt in opts:
10   if opt[0] == "-t": type = opt[1]
11   elif opt[0] == "-o": outdir = opt[1]
12   elif opt[0] == "-w": width = int(opt[1])
13   elif opt[0] == "-h": height = int(opt[1])

14 try:
15   tmp = pim.new("RGB", (1,1))
16   tmp.save("/dev/null", type)
17 except:
18   print >> sys.stderr, "ОШИБКА: формат %s не поддерживается" % type
19   sys.exit()

20 if not pattern: pattern = "."
21 filelist = []
22 for entry in pattern:
23   if entry[-1] == "/": entry = entry[:-1]
24   if os.path.isdir(entry): entry += "/*"
25   filelist.extend(glob.glob(entry))

26 for file in filelist:
27   if not os.path.isfile(file): continue
28   try:
29     im = pim.open(file)
30   except IOError: continue

31   if not (width and height):
32     if width: height = width * im.size[1] / im.size[0]
33     elif height: width = height * im.size[0] / im.size[1]
34     else:
35       width = im.size[0]
36           height = im.size[1]

37   resized = im.resize((width, height), pim.ANTIALIAS)
38   name, ext = os.path.splitext(file)
39   name = name.split("/")[-1]
40   outfile = "%s/%s.%s" % (outdir, name, type.lower())
41   resized.save(outfile)
42   print "%s (%s, %dx%d) -> %s (%s, %dx%d)" % (file,
43       im.format, im.size[0], im.size[1],
44           outfile, type, width, height)
```

нанных) аргументов. Наш скрипт будет ожидать во втором списке имена файлов (с указанием пути, если нужно), которые должны быть преобразованы. Пользователь может задать ширину результирующих изображений, их высоту или оба параметра сразу (если задан только один, второй вычисляется согласно пропорциям исходного файла, см. строки 31–36).

Поскольку в опции -t пользователь может ввести все, что угодно, в строках 14–19 мы проводим небольшую проверку – создаем изображение размером 1x1 и пытаемся записать его в **/dev/null** (чтобы нигде мусор не разводить) в указанном пользователем формате type (метод save() самостоятельно выполняет все нужные преобразования форматов). В случае неудачи сообщаем об этом пользователю (обратите внимание на синтаксис перенаправления вывода по дескриптору, отличному от stdout). Если же операция пройдет успешно – продолжаем.

Строки 20-25 отвечают за формирование списка файлов, соответствующих указанному шаблону. И, начиная с 26-й, мы приступаем собственно к обработке. Файлы, которые не могут быть открыты, например, текстовые, мы просто игнорируем – строка 30.

В строке 37 изменяем размер изображения (при этом создается новый объект). Флаг Image.ANTIALIAS задает способ интерполяции соседних точек (доступны и другие, например, LINEAR, NEAREST, BICUBIC). Наконец, рассчитав имя результирующего файла, выполняем сохранение. В данном случае мы не передаем методу save() второй параметр, указывающий формат файла – он будет определен автоматически по расширению.

Вот, собственно, и готово:

```
admin@toshiba:~/lxf/propy/l4/code$ ./conv2.py -t png -h 120 *.jpg
osf_t1.jpg (JPEG, 2272x1704) -> ./osf_t1.png (png, 160x120)
osf_t2.jpg (JPEG, 2272x1704) -> ./osf_t2.png (png, 160x120)
osf_t3.jpg (JPEG, 2272x1704) -> ./osf_t3.png (png, 160x120)
osf_t4.jpg (JPEG, 2272x1704) -> ./osf_t4.png (png, 160x120)
```

Небольшой совет: вместо метода resize() для создания миниатюр лучше использовать метод thumbnail() – он работает раза в два быстрее. Только имейте в виду, что thumbnail() изменяет текущий объект, т.е. исполняется «по месту», в то время как resize() создает копию объекта, оставляя оригинал в неприкосновенном виде (речь, естественно, об объекте в памяти – файл на диске не будет изменен до тех пор, пока вы явно не выполните метод save()).

Не конвертированием единым...

Естественно, возможности PIL этим не ограничиваются: она позволяет изменять цветовые схемы; применять к изображениям фильтры, такие как размывание, рельеф и т.д. (см. модуль *ImageFilter* и метод **filter()** модуля Image); вы можете смешивать различные изображения (методы **blend()** и **composite()**), поворачивать их на различные углы, смещать, вырезать из них фрагменты, накладывать простейшие графические элементы (такие как линии и дуги, см. модуль *ImageDraw*), и т.д.

Вот так, например, можно в два раза усилить «красную» составляющую исходного изображения:

ЛИСТИНГ MAXIRED.PY

```
#!/usr/bin/Python
# -*- coding: utf-8 -*-
```

»

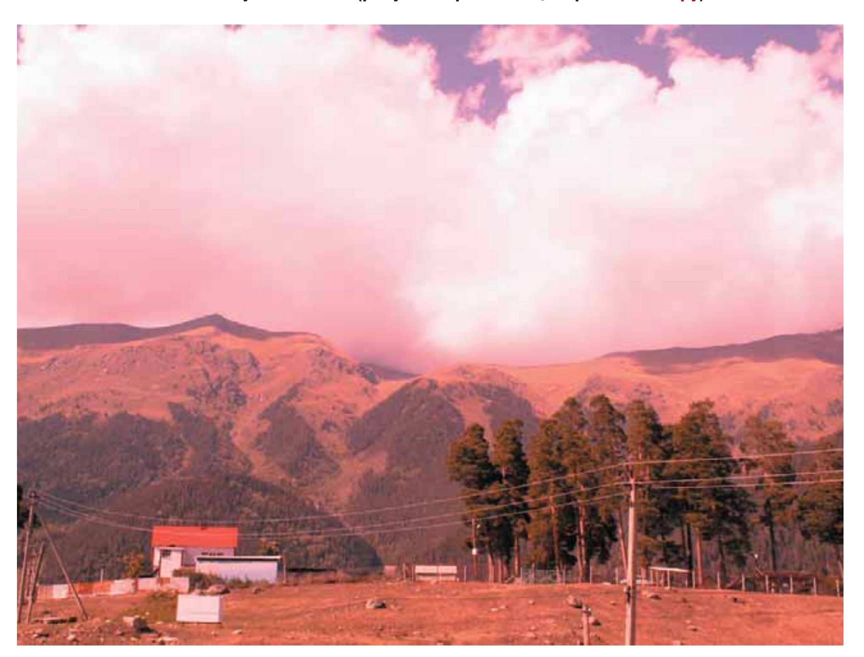
```
import os, sys, Image

infile = sys.argv[1]
im = Image.open(infile)
if im.mode == “RGB”:
  imR, imG, imB = im.split()
  imR = imR.point(lambda pixel: pixel * 2)
  outfile = Image.merge(“RGB”, (imR, imG, imB))
  outfile.save(“%s-red%s” % os.path.splitext(infile))
else:
  print >> sys.stderr, “ОШИБКА: файл не RGB”
```

Здесь методом split() мы раскладываем исходный объект на «цветовые» составляющие (если атрибут объекта, mode, подтверждает, что загруженный файл является RGB-изображением), методом point() применяем указанную функцию (в нашем случае - умножение на два) к каждому пикселу «красного» объекта, и с помощью merge() объединяем все снова в одно изображение.



Вот так, десятком строк, мирный утренний пейзаж превращается в буйный закат (результат работы сценария maxired.py).



Листинг «playit.py»

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import pymedia.audio.acodec as acodec
import pymedia.audio.sound as sound
import pymedia.muxer as muxer
import time

class playit:
    def __init__(self, file2play, rateCf=1):
            self.PLAY_FLG = 1
            self.file2play = file2play
            self.rateCf = rateCf

    def playit(self):
            dm = muxer.Demuxer(self.file2play.split(‘.’)[-
1].lower())
            f = open(self.file2play, ‘rb’)
            s = f.read(32000)
            dm.parse(s)
            dec = acodec.Decoder({‘id’:dm.streams[0][‘id’]})
            r = dec.decode(s)

            print ‘Play “%s”:’ % self.file2play
            print ‘   Bitrate:       %d’ %r.bitrate
            print ‘   Sample Rate:   %d’ % r.sample_rate
            print ‘        RateCf:   %.2f’ % self.rateCf
            print ‘   Channels:      %d’ % r.channels

            snd = sound.Output(int(r.sample_rate * self.rateCf),
                                  r.channels,
                                  sound.AFMT_S16_LE)

            while len(s) > 0:
                    if r: snd.play(r.data)
                    s = f.read(512)
                    r = dec.decode(s)

            while snd.isPlaying(): time.sleep(1)

            self.PLAY_FLG = 0
```

Нужно заметить, что библиотека PIL предназначена для обработки изображений, а не для их вывода. Если вы хотите вставить фотографию в ваше графическое приложение, для этого следует использовать средства соответствующего модуля. Например, в *Tkinter* для этого предназначены классы BitmapImage и PhotoImage. Есть, конечно, и в PIL метод show(), но он просто сохраняет объект во временный файл и отдает его внешней программе просмотра изображений (*xv* под Unix/Linux, *Paint* в Windows), поэтому пригоден он разве что для отладочных целей, чтобы по ходу разработки скрипта можно было «на лету» контролировать промежуточные результаты.

Нам песня строить и жить помогает...

Следующая библиотека, которую мы затронем в этом уроке – *PyMedia*. Скачать исходный код можно с сайта www.pymedia.org (там же вы найдете и документацию, и архив с примерами). Установка выполняется, в принципе, достаточно просто:

```
admin@toshiba:~$ tar xzvf PyMedia-1.3.7.3.tar.gz
admin@toshiba:~$ cd PyMedia-1.3.7.3
admin@toshiba:~/PyMedia-1.3.7.3$ sudo Python setup.py install
```

Однако, среди зависимостей – *alsa*, *ogg*, *vorbis*, *faad*, *mp3lame* (причем для инсталляции нужны и заголовочные файлы, т.е. может потребоваться доустановить соответствующие dev-пакеты). Также

потребуется пакет *Python-dev*. Так что, если вам посчастливится найти для своего дистрибутива двоичный пакет библиотеки (в частности, deb-пакет, правда, не самой новой версии, можно найти здесь: **http://prdownloads.sourceforge.net/PyMedia/PyMedia_1.3.5_i686-py2.4.deb?download**), то рекомендую воспользоваться им.

На листинге «*playit.py*» представлен код класса, задача которого – воспроизвести звуковой файл. Первыми строками (3–6) подключаем нужные нам модули (иерархия здесь довольно сложная, так что без документации не обойтись; ну и про примеры не забывайте).

Сначала мы должны определить идентификатор кодека, который подойдет для воспроизведения того или иного файла. Для этого мы создаем объект Demuxer, которому передается в качестве параметра расширение файла (строка 13). Этот объект мы используем для того, чтобы декодировать заголовок звукового файла (в строках 14–16 мы считываем первые 32000 байт файла и передаем их методу parse()). Теперь мы можем определить идентификатор нужного кодека по ключу 'id' в словаре dm.streams[0] (строка 17; можно в качестве параметра передать и весь словарь), и используем этот кодек для декодирования уже считанных данных. Поскольку здесь, помимо всего прочего, присутствует и служебная информация, то после декодирования мы получаем возможность вывести кое-что из этого на экран (строки 19–23; но это только в тестовых целях – в классах, которые могут использоваться в различных программах, оператору print, конечно, не место).

В строке 24 создается объект Output, метод play() которого будет использоваться для воспроизведения (строка 28). Изменив параметр rateCf, можно ускорить или замедлить проигрывание файла (по умолчанию используется значение 1). Наконец, в цикле (строки 27–30) мы воспроизводим уже считанный фрагмент (переменная s) и готовим следующий. Как только мы считаем и передим методу play() весь файл, в строке 31 начнется ожидание конца воспроизведения (чтобы не сильно нагружать процессор, проверку выполняем один раз в секунду). Ожидание нужно, поскольку play() является асинхронным методом, который возвращает управление сценарию до того, как воспроизведение фрагмента на самом деле завершится. Благодаря этому мы можем спокойно обрабатывать следующий фрагмент, не беспокоясь о паузах при воспроизведении.

Для использования класса playit нужно создать соответствующий объект, передав ему имя файла и, при желании, коэффициент, корректирующий скорость воспроизведения. Само воспроизведение запускается методом playit(). Причем библиотека *PyMedia* вполне позволяет поигрывать одновременно несколько файлов - в качестве примера рассмотрим, как можно получить простейший эффект объемного звука:

Листинг echo.py

```
#!/usr/bin/Python

from playit import playit
import threading, time

mp3 = 'Aria.shtil.mp3'
ECHO = 0.1

f1 = playit(mp3)
f1.rateCf = 1
ht1 = threading._start_new_thread(f1.playit, ())

if ECHO:
        time.sleep(ECHO)

        f2 = playit(mp3)
        f2.rateCf = 1
        ht2 = threading._start_new_thread(f2.playit, ())

while(f1.PLAY_FLG or ECHO and f2.PLAY_FLG):
        time.sleep(1)
```

Листинг wav2.py

```
1   #!/usr/bin/python
2   # -*- coding: utf-8 -*-

3   import wave

4   def sum(a, b):
5      return “%c%c” % (a, b)

6   rf = wave.open(“new.wav”, “rb”)
7   print “Количество фреймов:    %d “ % rf.getnframes()
8   print “Частота дискретизации:  %d” % rf.getframerate()
9   print “Точность дискретизации: %d” % rf.getsampwidth()
10  print “Количество каналов:     %d” % rf.getnchannels()
11  print “Тип сжатия:             %s” % rf.getcomptype()

12  total = rf.getnframes() * rf.getnchannels() * rf.getsampwidth()
13  normal = rf.readframes(total)
14  reverse = ‘’
15  if rf.getsampwidth() == 2:
16     n2 = normal[-2::-2]
17     n1 = normal[-1::-2]
18     for i in map(sum, n0, n1):
19        reverse += i
20  else:
21     reverse = normal[-1::-1]

22  wf = wave.open(“new2.wav”, “wb”)
23  wf.setnchannels(rf.getnchannels())
24  wf.setsampwidth(rf.getsampwidth())
25  wf.setframerate(rf.getframerate())
26  wf.writeframes(reverse)
```

В этом сценарии мы запускаем два потока (если значение переменной ECHO отлично от нуля). В каждом из потоков проигрывается один и тот же файл, но перед запуском второго делается небольшая пауза (в примере – 0,1 секунды). В результате получается довольно интересное звучание. Кстати, обратите внимание на то, как мы запустили потоки – вместо того, чтобы создавать соответствующий класс и переопределять в нем метод run(), был напрямую вызван «внутренний» метод _start_new_thread(), которому передается функция или метод, которые должны быть выполнены в потоке.

Помимо собственно воспроизведения звука, *PyMedia* предоставляет средства для работы с видео-файлами, ее можно использовать для конвертирования исходного файла в другие форматы, для редактирования мета-данных и самого звука, и т.д.

Ссылки по теме

www.python.org – официальный сайт проекта
www.python.ru – российский сайт почитателей этого языка
http://zope.net.ru – сайт российской группы пользователей среды Zope (разработанный на Python сервер web-приложений); здесь есть информация и о Python
www.pythonware.com – сайт разработчиков PIL; помимо самой библиотеки, можно найти и другие интересные вещи
www.awaretek.com/plf.html – страничка, посвящённая изучению языка Python
http://py.vaults.ca/apyllo.py – множество примеров программ на Python, от работы с базами данных и графикой до игр

Малышка wave

Кстати, раз уж зашла речь об обработке звука, то и в стандартной поставке *Python* есть несколько модулей для выполнения простейших операций. Например, модуль wave позволяет редактировать (и даже создавать с нуля, если вы сильны в математике) звуковые файлы в формате WAV. На листинге **wav2.py** представлен сценарий, с помощью которого можно «развернуть» wav-файл в обратном направлении:

Открыв wav-файл (строка 6), мы можем получить некоторую информацию о нем (строки 7-11). Считываем данные в переменную normal (нормальная последовательность). «Сырой» wav-файл (т.е. без компрессии; подавляющее большинство таковыми и является) представляет собой набор «фреймов», или «отсчетов», то есть значений амплитуды сигнала в данный момент времени. Частота дискретизации определяет, сколько фреймов будут формировать одну секунду звучания. Точность дискретизации показывает, сколько байт используется для хранения одного фрейма (значение 1 соответствует 8-битному звуку, 2 – 16-битному).

Для того чтобы развернуть наш файл, нужно «реверсировать» считанную нормальную последовательность. Это можно легко выполнить с помощью списковых включений (как показано в строке 21). Но если мы работаем с 16-битным файлом, то нужно позаботиться о сохранении порядка байтов во фрейме, иначе получим лишь шум. Для этого в строках 16 и 17 мы формируем две «подпоследовательности», в одну из которых попадают младшие байты фреймов, во вторую – старшие. С помощью функции map(), которая применяет функцию, указанную первым параметром (в нашем примере это sum(), описанная в строках 4–5), к каждому элементу последовательностей, переданных вторым и третьим параметрами, мы соединяем полученные полуфреймы воедино. Поскольку на выходе map() получается список, то нам нужно его вручную «собрать» снова в строку (стр. 18–19).

Нужно заметить, что map() отличается очень высокой скоростью обработки последовательностей. Если собирать результирующую строку «вручную» (например, в цикле while), то эта процедура может затянуться на десятки секунд даже для не очень большого файла, в то время как map() выполняет эту операцию почти мгновенно.

Наконец, в строках 22-26 мы открываем новый файл на запись, устанавливаем его параметры (в соответствии с исходными) и записываем содержимое. Теперь можете и сами поиграть в «АПОП»!

Только вперед!

На этом мы завершаем нашу серию уроков. Благодаря тому, что разработчикам *Python* удалось сделать этот язык простым и в то же время удивительно мощным и расширяемым, с его помощью можно эффективно решать самые различные задачи – дополнительная сложность будет проявляться лишь там, где это действительно необходимо, в то время как простые задачи сохранят предельную простоту. И если вдруг для какого-то проекта стандартных средств окажется недостаточно, то наверняка вам поможет какая-нибудь из сторонних библиотек. Главное – не сдаваться!

Удачи! LXF



Базовые

ЧАСТЬ 2: Мы говорим на русском языке и пользуемся операционной системой GNU/Linux – подобное на нашей планете встречается не так уж часто. Хотите стать ещё более элитарным подмножеством? **Евгений Балдин** знает уникальный рецепт – используйте LaTeX!



«Командная логика»

Щенок должен понять,
что быстрое и чёткое выполнение команд хозяина
всегда вознаграждается лакомством или лаской.

Для набора кода в *LaTeX* знание английского языка приходится очень кстати. Основных команд немного – их можно запомнить и так, но для совершенствования английский необходим, хотя бы для чтения документации к пакетам. Названия у команд, как правило, вполне осмысленны, что очень помогает при поиске чего-нибудь необходимого в алфавитном указателе. «Правильный» тестовый редактор тоже не является лишним.



Спецсимволы

Не все символы равноправны. За частью символов в *LaTeX* зарезервированы специальные значения.

Это: «\», «$», «%», «_», «{», «}», «&», «#», «^» и «~». В процессе изложения их роль будет, со временем, раскрыта.

Чтобы отобразить эти символы при печати необходимы дополнительные усилия. Предыдущий параграф в текстовом редакторе выглядел бы так:

```
Это: <<\textbackslash>>, <<\$>>, <<\%>>, <<\_>>[1] , <<\{>>, <<\}>>,
<<\&>>, <<\#>>, <<\^{}>> и <<\~{}>>. В процессе изложения их роль
будет раскрыта.
```

Группировка

Группировка осуществляется с помощью фигурных скобок: {группа}. Фигурные скобки при печати не отображаются.

Сложные конструкции, которые имеют открывающую и закрывающую команды (например, окружения), тоже группируют текст.

Построение команды

Команды в *LaTeX* начинаются с символа \ (backslash или обратная косая черта) и продолжаются комбинацией, состоящей только из стандартных латинских букв[2]. Команды завершаются пробелом, цифрой или не латинской буквой. Все пробельные символы после команды игнорируются. Для того, чтобы пробел после команды не игнорировался, достаточно вставить «пустую группу»: \command{}. Например, чтобы лого *LaTeX* (команда \LaTeX) не слилось со следующим за ним словом, следует написать \LaTeX{}.

Аргументы

Командам *LaTeX* могут передаваться внешние аргументы:

```
\command[param1][param2]{param3}{param4}
```

В квадратные скобки заключаются необязательные параметры (param1 и param2), а в фигурные – обязательные (param3 и param4).

Некоторые из команд *LaTeX* влияют только на свои аргументы. Например, команда

```
\textbf{текст}
```

выводит слово текст жирным шрифтом.

Декларативные команды

Часть команд *LaTeX* являются своеобразными переключателями режимов.

Область действия декларативной команды может ограничиваться логической группой или единицей структуры печатного документа, например, страницей.

```
%дальнейший текст будет печататься жирным шрифтом
\bfseries
%убрать заголовки и нумерацию для текущей страницы
\thispagestyle{empty}
```

[1] Можно воспользоваться пакетом *underscore* – в этом случае необходимость экранировать знак «_» в текстовом режиме отпадает.

[2] Это верно для стандартной кириллизации. Существуют варианты – например, русификация Шеня – где русские буквы также могут входить в имена команд.

Месяц назад Мы познакомились с «клеем» и «боксами».

элементы

Окружения

Сложные конструкции, которые имеют открывающую и закрывающую команды вида \begin{имя} и \end{имя}, называют окружениями. Вместо слова «имя» подставляется название соответствующего окружения

```
\begin{center}
Это строка будет отцентрирована
\end{center}
```

Окружения могут вкладываться друг в друга, как матрёшки, но их области действия не могут перекрываться.

Логика документа

Вид документа определяется выбором класса и стилей. Хорошим приёмом является наличие личных готовых шаблонов с уже предопределёнными предпочтениями, которые могут меняться по мере развития документа.

Структура файла LaTeX

Текстовый tex-файл состоит из двух частей: заголовка или преамбулы и собственно самого текста, и выглядит примерно следующим образом:

```
%---начало заголовка---
%выбор класса документа, например article или book
\documentclass{…}
%минимальная кириллизация
\usepackage[koi8-r]{inputenc}
\usepackage[english,russian]{babel}
\usepackage{indentfirst}
%загрузка пакетов по выбору
\usepackage{…}
  …
%определение своих команд или переопределение уже
существующих
\newcommand{\mycommand}{…}
\renewcommand{\oldcommand}{…}
  …
%локальные настройки
  …
%---конец заголовка---
\begin{document}
%тело документа
\end{document}
```

Первым делом с помощью инструкции \documentclass осуществляется выбор класса документа. Далее загружаются стилевые пакеты.

Для того, чтобы можно было набирать русский текст, необходимо с помощью пакета *inputenc* указать кодировку текстового файла, например, koi8-r. Далее нужно подключить пакет *babel*, который отвечает за локализацию, в частности, за переносы и «национальные особенности» набора. Например, при включении русского языка доопределяется символ номера № (\No), символ параграфа § (\S) и многое другое. Для формирования отступа или красной строки у первого параграфа, как это принято в России, необходимо загрузить пакет *indentfirst*. По идее, это должно относиться к «национальным особенностям», но в *babel* по умолчанию не подгружается.

К вопросу о переносах

Пожалуй, единственное, что, возможно, потребует настройки в свежеустановленном дистрибутиве *LaTeX*, это включение переносов. Этой проблемы заведомо не возникает при установке *TeX Live* и в большинстве современных дистрибутивах GNU/Linux, но всякое бывает.

Для установки переносов можно воспользоваться утилитой *texconfig*[3]. После запуска программы следует выбрать меню HYPHENATION, а затем меню LaTeX. Далее будет предложено отредактировать[4] файл переносов **language.dat**. Обычно достаточно таких настроек:

```
english hyphen.tex
russian ruhyphen.tex
```

Всё остальное по желанию можно закомментировать. По умолчанию, когда включаются переносы для определённого языка, все остальные правила переносов отключаются. Но в случае английского и русского языков это можно обойти, воспользовавшись русско-английской таблицей переносов:

```
ruseng ruenhyph.tex
=russian
=english
```

Следует понимать, что подобная настройка с точки зрения философии *LaTeX* не совсем корректна. Для гарантированно одинакового результата компиляции независимо от платформы лучше поступиться некоторыми удобствами.

Класс документа

С помощью обязательной инструкции \documentclass[опции]{класс} можно указать, к какому классу будет относиться рабочий текст.

Класс документа следует выбирать в зависимости от того, что должен представлять из себя текст. Для начала можно остановиться на стандартном классе article. Этот класс разработан специально для статей и небольших отчётов. Для отчётов побольше можно использовать класс report, а для книг – класс book.

Перечисленные стандартные классы сложились очень давно и многие производные классы документов основаны на них. Как следствие, стандартные базовые классы абсолютно статичны, поэтому для специализированных вещей используются свои классы. Например, для научных статей популярны различные модификации revtex4. Вячеслав Фёдоров разработал класс eskd (стандарт ЕСКД), который можно найти на CTAN или в стандартной поставке *TeX Live*. Те, кто недоволен большими полями в стандартных *LaTeX*-классах, могут обратить внимание на набор классов *KOMA-script* (scrartcl, scrreprt и scrbook вместо article, report и book, соответственно).

Поначалу в выбранном классе ничего менять не следует. То, что кажется с непривычки неудобным, на самом деле может улучшать восприятие печатной копии. Например, относительно узкая ширина текста в стандартных классах (следствие больших полей) позволяет при прочтении охватывать взглядом всю строку целиком, что увеличивает скорость чтения.

---

[3] *texconfig* – это простенькое dialog-подобное консольное приложение. Некоторые настройки могут потребовать привилегий суперпользователя. После изменения настроек через *texconfig* автоматически пересоздаются необходимые форматные файлы. В противном случае может потребоваться сделать это вручную, например, с помощью инструкции вида texconfig init.

[4] Редактор можно определить с помощью переменной окружения $EDITOR. Если переменная не определена, то вызывается редактор *vi*. В случае отсутствия опыта работы с *vi* следует выйти из него с помощью последовательности :q и настроить переменную окружения.

» Как правило, разумные модификации можно осуществить через параметры в команде выбора класса, например:

```
\documentclass[a4paper,12pt,oneside]{scrbook}
```

a4paper – размер листа бумаги (можно выбрать другой стандарт, например, a5paper), 12pt – базовый размер шрифта (в стандартных классах доступны размеры в 10pt и 11pt), а oneside – односторонняя печать (удобнее при просмотре электронной версии).

В заключение хотелось бы отметить набор классов NCC, активно разрабатываемых А.И.Роженко. Класс ncc позиционируется автором как «русскоязычная статья». Класс можно взять на CTAN, он присутствует и в стандартной поставке *TeX Live*.

Стили

Стилевой файл **(.sty)** или пакет представляет из себя набор макросов и определений, созданных для решения какой-то определённой задачи. Для подключения стилевого файла используется команда \usepackage[опции]{стиль}.

Основное отличие классов от пакетов в том, что один документ может включать один класс и сколько угодно стилевых пакетов. Фактически, на любую задачу в *LaTeX* находится ответ в виде соответствующего пакета. В стандартной поставке *TeX Live* присутствует свыше двух тысяч **.sty**-файлов, кроме того, ничего не мешает создать свой, оптимизированный под локальные задачи.

К вопросу о кириллизации

Чтобы кириллизовать *LaTeX*, необходимы шрифты. Благодаря Ольге Лапко на свете есть шрифты семейства lh, которые отлично согласуются с базовыми шрифтами Computer Modern. Мало иметь просто кириллические буквы – надо, чтобы их начертания соответствовали и другим шрифтам, в том числе, математическим. В 2001 году Владимир Волович проделал огромную работу по переводу METAFONT-шрифтов в формат Type1, что теперь позволяет создавать не только хорошие печатные копии, но и вполне качественные электронные pdf-версии документов.

За перевод из кодировки файла во внутреннюю кодировку *LaTeX* отвечает пакет *inputenc*. В качестве опции при загрузке с ним передаётся текущая восьмибитная кодовая страница документа. Для кириллицы могут оказаться интересны следующие варианты: koi8-r, koi8-u, cp866, cp1251 и 8859-5. Собственно говоря, всё. Единственное неудобство, которое возникает из-за этого, заключается в том, что сообщения об ошибке *LaTeX* выдаёт в своей внутренней T2A кодировке[5]. Для исправления этого неудобства можно воспользоваться простейшим фильтром. Для начало его надо собрать:

```
> locate t2filter.c
{TEXMF}/texmf-dist/doc/generic/t2/etc/t2filter.c
> cd {TEXMF}/texmf-dist/doc/generic/t2/etc/
> gcc -Wall -O2 -s -o ~/bin/t2filter t2filter.c
> LaTeX {файл}.tex | t2filter
```

Тело документа

Всё, что заключено внутри окружения document, является телом документа. Если у вас есть какие-то куски текста, которые печатать не хочется, а выкинуть жалко, то их достаточно вынести в конец за инструкцию \end{document}.

Логика набора

Объявление в газете: Ищу работу машинистки.
Печатаю со скоростью 4000 знаков в минуту.
Правда, такая белиберда получается!

Мало открыть файл в текстовом редакторе и начать набирать. Нажимать на клавиши надо осмысленно.

Печатаем текст

При наборе книги/статьи/заметки основное вовсе не команды, а сам текст. Правила очень просты.

Комментарии

Всё, что следует за знаком «**%**» включительно, является комментарием.

Большие закомментированные сегменты мешают работать с основным текстом, и поэтому их следует исключать из рабочего файла. Но при желании можно воспользоваться окружением comment из пакета *verbatim*.

Разделение слов

Пробельные символы используются в *LaTeX* для разделения слов. Пробелы в начале строки игнорируются. Символ перевода строки так же воспринимается как пробел. Если в конце строки сразу за последним словом вставить знак комментария:

```
экранировка перевода стр%
   оки
```

то разделения слов не происходит. Иногда этот приём может оказаться полезным.

Разделение абзацев

Для того, чтобы начать новый абзац, необходимо оставить пустую строку:

```
текущий абзац закончился

следующий абзац начался
```

Число пустых строк между абзацами не имеет значения.

Пунктуация

Напечатанный текст обезличивается. Нет эмоций – только буквы. Единственное, что остаётся – это знаки пунктуации и, возможно, смайлики ☺.

Запятую, точку, точку с запятой, двоеточие, многоточие, скобки, кавычки, восклицательный и вопросительные знаки следует «прижимать» к словам. Не надо оставлять пробелов, а то *LaTeX* «подумает», что так и надо.

Пробелы

Расстояние между словами *LaTeX* выбирает по своему усмотрению для максимально равномерного заполнения страницы. Но иногда необходимо указать размер пробела вручную:

- «**~**» – неразрывный пробел, т.е. по этому пробелу не производится перенос на другую строку,
- «**\,**» – маленький нерастяжимый пробел,
- «**\ **» – нормальный нерастяжимый пробел.

```
В основном, указывать размеры пробелов надо в случае набора
каких-либо сокращений, например, так следует набирать ФИО:
Ф.\,А.~Милия "--- негоже <<отрывать>> ИО от Ф\@. Ещё примеры:
т.\,е., г.~Новосибирск, рис.~1 и~т.\,д.\ и~т.\,п.
```

LaTeX считает, что после точки предложение заканчивается, если эта точка стоит не после заглавной буквы. Растяжимость пробелов между предложениями и между словами существенно разная. Поэтому если точка случается в середине предложения, то после неё следует явно вставить пробел «**\ **» или неразрывный пробел «**~**». Может случиться, что точка следует сразу за заглавной буквой и означает именно конец предложения (как в примере происходит с буквой Ф). Для этого перед такой точкой следует добавить коррекцию в виде команды «**\@**».

Дефисы, минусы и тире

В издательских системах, основанных на TeX'е, различают дефис (hyphen), короткое тире (en-dash), длинное тире (em-dash) и знак минуса.

Чтобы получить на печати дефис, короткое или длинное тире, надо набрать один, два или три знака «**-**», соответственно.

[5] Расположение букв похоже на таковое в кодировке **cp1251** – это чистая случайность.

При подключении пакета *babel* с опцией russian появляются дополнительные команды, позволяющие более строго следовать русским печатным традициям.

```
Дефис используют в составных словах (кто-то, где-нибудь),
короткое тире рекомендуется для указания диапазона чисел (10--15,
2001--2006), длинное тире означает обычное тире (\LaTeX{} “--- это
круто), минус может существовать только в формулах ($a-b=c$).
Пакет \texttt{babel} вводит дополнительные команды для написания
тире. Для двойных или парных фамилий следует использовать
конструкцию “--~, например, уравнение Менделеева”--~Клапейрона,
композитор Римский”--~Корсаков. Чтобы длинное тире не
отрывалось от предыдущего слова и вокруг него создавались
правильные пробелы вместо --- следует употреблять “---, т.\,е. к
трём тире надо добавить двойную кавычку. Прямая речь должна
начинаться с команды “--*:
“--* Я сказал.
```

Правила могут показаться немного запутанными, но к ним быстро привыкаешь, и они того стоит.

Переносы

В большинстве случаев *LaTeX* грамотно переносит слова, но в случае сложных слов, которые пишутся через дефис, перенос происходит только по дефису. Аналогично проблемы возникают, когда слово частично состоит из английских букв, а частично из кириллицы.

```
Прямо в тексте перенос можно указать с помощью команды \-,
например: дель\-та-функ\-ция, \TeXно\-ло\-гия.
При наличии русского языка в \texttt{babel} вместо дефиса в
сложном слове можно поставить команду “=, например,
дельта”=функция. В этом случае переносы будут сделаны корректно
без подсказки.
```

Для часто упоминаемых слов можно задать шаблон переноса с помощью команды:

```
\hyphenation{образ-цы пе-ре-но-са дель-та=-функ-ция}
```

Обычно, образцы переноса лучше определять в заголовке документа. Следует понимать, что образцы автоматически не склоняются, поэтому надо предусмотреть всевозможные варианты окончаний.

С помощью команды \hyphenation можно запретить перенос слова в нежелательных местах, просто не указав место разрыва. В тексте запрет переноса можно оформить с помощью инструкции \mbox{нет переноса}.

Многоточие

Многоточие печатается с помощью команды \ldots. Если многоточие идёт после точки, то необходимо вставить неразрывный пробел **~**.

Ударение

В русском языке длительность ударного гласного примерно в 1.5–2 раза длиннее безударного. Если ударение поставить не в том месте, то слово будет звучать совсем по другому.

```
В корне \textbf{зар-} “--- \textbf{зор-} под ударением пишется  гласная
в соответствии с произношением, без ударения “--- \textbf{а}.

\emph{Исключения:} зор\'янка, озар\'ять.
```

„Лапки” и «Ёлочки»

В пакете *babel*, кроме всего прочего, определены традиционные русские кавычки.

Если в начале или в конце текста встречаются внутренние и внешние кавычки, то они должны различаться между собой рисунком.

```
Он сказал: <<А пойду-ка я и подпишусь на ,,Linux Format''>>.
```

Структурная логика

LaTeX ориентирован на логическую разметку документа. Можно конечно «сказать», что данный кусок текста следует напечатать 20-м кеглем, выровнять по левому краю и сделать отступ после него в два интервала, но проще указать, что это заголовок раздела.

Титульный лист

Создания титульного листа это отдельная задача, в которой визуальная составляющая обычно превалирует над структурной. В этом случае следует воспользоваться окружением titlepage. При инициализации этого окружения создаётся чистая страница, которой присваивается номер один, а содержание этой страницы полностью определяется фантазией автора. Но, в любом случае, это следует делать после написания самого текста. Обычно достаточно стандартного заголовка:

```
\title{\LaTeX, Unix и русский стиль}
\author{E.\,M.~Балдин\thanks{e-mail: E.M.Baldin@inp.nsk.su}}
\date{2006}
\maketitle
```

Команда \maketitle создаёт стандартный титульный заголовок, используя информацию о названии документа (\title), авторе (\author) и дате написания текста (\date). Команда \thanks правильным образом позволяет оформить подстрочное примечание на титульной странице. Если авторов более чем один, то их можно перечислять, разделяя командой \and – в этом случае список авторов печатается в виде таблицы.

В статьях (производные от класса article) вслед за заголовком следует обязательная аннотация, которая оформляется с помощью окружения abstract.

За буковку «ё» замолвите слово

Пара слов в поддержку буквы «ё». Эту букву незаслуженно забывают при наборе текстов. Более того, некоторые деятели ратуют за её полное упразднение. Однако, коль уж вам довелось узнать русский язык, то говорить и писать на нём следует правильно. Наличие буквы «ё» в тексте значительно облегчает процесс чтения. Особенно это актуально при быстром чтении.

Поставь букву «ё» на место её!

Для проверки правописания следует использовать словарь Александра Лебедева. Этот словарь построен на основе словаря русского языка для ispell, первоначально составленного Нилом Далтоном (Neal Dalton) в 1992 г. После тщательной проверки и исправления примерно 4000 ошибок в словаре Нила Далтона, в словарь были добавлены отсутствовавшие в нём правила образования форм существительных, прилагательных, причастий, наречий, изменены правила формирования окончаний глаголов, так что affix-файл можно считать переписанным заново. Одновременно в словарь было добавлено большое число слов.

Отличительной чертой данного словаря является то, что в него включена полноценная поддержка буквы «ё». В современных дистрибутивах GNU/Linux словарь Александра Лебедева является основным русским словарём для ispell. К сожалению, в подавляющем большинстве случаев словарь «собран» без поддержки «ё». Авторы пакета ошибочно предполагают, что «е» и «ё» – это одинаковые буквы. При проверке правильными являются такие слова как «ежик», «елка» и тому подобное. Правильный выход: убедить автора/самому стать автором пакета. Неправильный, но гораздо более простой: локально пересобрать пакет с полноценной поддержкой «ё».

Словарь постоянно совершенствуется, дополняется и корректируется. Последнюю версию словаря можно найти на авторской страничке Александра Лебедева: http://semiconductors.phys.msu.su/~swan/orthography.html

TeX-логослово

Когда Д.Э. Кнут создавал TeX – он много думал. Причём думал не только об алгоритмах и коде. В частности, он нашёл время подумать о том, как назвать своё произведение. TeX читается как «тех». Последняя буква – вовсе не английская буква «икс», а греческая «хи». Также он продумал и правила изображения этого названия. С тех пор в TeX-сообществе возникла мода на создание TeX-лого.

TEX	\TeX
LATEX	\LaTeX или \LATEX
LATEX 2ε	\LaTeXe
AMS-TEX	\AMSTeX или \AmSTeX
METAFONT	\METAFONT или \MF
BibTEX	\BIBTeX или \BibTeX
NCC	\NCC

Распространённые TeX-лого. Команда \NCC определена в пакете *ncclatex*. Остальные команды заведомо определены в пакете texnames.

Секционирование

Часто бывает полезно сразу же за титульной страницей вывести оглавление с помощью команды \tableofcontents. Для этого в тексте должно присутствовать логическое разбиение на разделы.

```
\subsection{Секционирование}
\label{sec:base:sec}

Часто бывает \ldots
```

Команды секционирования образуют строгую иерархию. Самыми старшими по «званию» являются разделы \part{Часть} и \chapter{Глава}. Это большие куски текста, и, соответственно, их применение обосновано только в книгах, поэтому они не определены в классах производных от article и report, зато определены в классе book.

Далее по старшинству следуют:

```
\section{Раздел}
\label{ex:section}

\subsection{Подраздел}
\label{ex:subsection}

\subsubsection[<<Подподраздел>>]{Что-то более мелкое чем
подраздел}
\label{ex:subsubsection}

\paragraph{Параграф}
\label{ex:paragraph}

\subparagraph{Подпараграф}
\label{ex:subparagraph}
```

Если воспользоваться необязательным параметром команды секционирования, его значение заместит основной заголовок при печати оглавления и создании колонтитулов.

Команды секционирования печатают заголовок необходимым шрифтом и нумеруют раздел. Если нет желания, чтобы название раздела попало в оглавление, и надобности в нумерации нет, то к команде секционирования следует добавить символ «*», например:

```
\section*{Приложение}
```

Перекрёстные ссылки

Одной из основных причин, по которой *LaTeX* вытеснил обычный TeX из текстовых редакторов TeXников, является механизм нумерации и создания ссылок.

Чтобы сослаться на раздел, в нём необходимо оставить метку \label{метка}. А затем можно использовать команды:

```
В разделе~\ref{ex:section} на странице~\pageref{ex:section} \ldots
```

Когда ссылки идут через метку, то номер раздела и номер страницы определяется *LaTeX* автоматически. Причём автоматическая нумерация свойственна не только командам секционирования – точно так же можно ссылаться на формулы, таблицы, картинки и листинги программ. Для этого необходимо оставить метку \label в соответствующем окружении.

Сложные документы

Всё можно хранить в одном файле – это ничему не противоречит. Более того, само понятие «файл» для пользователя не так уж и необходимо. Но уж если файл есть, то почему бы не разделить большой текст на несколько частично независимых кусков?

С помощью команды \input{имя файла} можно вставлять в документ другой tex-файл. *LaTeX* просто добавляет содержимое по месту команды, считывая файл либо до конца, либо до первой встретившейся инструкции \endinput. Совершенно не важно, в какой части документа встречается \input. Бывает довольно удобно вынести преамбулу в отдельный файл. В имени файла можно опустить расширение **.tex**.

Для включения текста можно применить другой способ:

```
\input{preheader}
\includeonly{
% intro,
 base,
% presentation
}
\begin{document}
%введение
\include{intro}
%базовые команды
\include{base}
%Презентация
\include{presentation}
\end{document}
```

Декларация \include позволяет включить только tex-файл (при написании имени расширение **.tex** опускается). В преамбуле с помощью команды \includeonly можно перечислить, какие части надо подключить при текущей сборке. При этом сохраняется правильная нумерация страниц и можно сослаться на не включённые в эту сборку разделы. Это довольно актуально в случае больших текстов в процессе их создания, так как значительно ускоряет компиляцию. LXF



Через месяц Изучим математику на уровне школьной программы.

Сказка Java

ЧАСТЬ 1: От кофеварок до промышленных серверов и биллинговых систем – Java можно встретить буквально везде. Начните изучение этой технологии вместе с **Антоном Черноусовым**.



Начиная ряд статей о программировании на объектно-ориентированном языке Java, хотел бы выразить свое мнение относительно корпоративного рынка разработки программного обеспечения, на котором в данный момент борются за лидерство два гиганта, две платформы: J2EE и .NET.

В основе платформы J2EE лежит язык программирования Java – приложения, созданные с его помощью, являются кроссплатформенными, сам язык, средства разработки и технология активно поддерживаются многими корпорациями: Sun, Google, IBM, Oracle, BEA. Платформа пользуется большой популярностью и среди компаний, ориентированных на *OpenSource*, таких как *Apache Foundation*. В противовес J2EE, корпорация Microsoft предлагает свою технологию .NET, которая является колоссом, построенным на мощи программного обеспечения Microsoft.

Сегодня еще нельзя сказать, какая технология станет доминирующей, и, возможно, от вас зависит скорость распространения современной кроссплатформенной технологии создания приложений J2EE и языка программирования Java.

Предложенные вашему вниманию уроки позволят овладеть основными понятиями ООЯ Java и создавать различные приложения с его помощью. Рассмотрев основы языка, мы сможем изучить типовые решения (паттерны) в программировании и приступить к изучению J2EE.

В первом уроке мы создадим свое первое приложение и разберемся с основными понятиями: инкапсуляция, наследование и полиморфизм. Для выполнения этого урока нам понадобится пакет Java SDK 1.5, текстовый редактор и немного терпения.

Героический объект

Приступая к этой серии материалов, я вспомнил, что, как правило, изучение языков программирования начинается с написания приложения «Hello World!», но такой прием настолько изъезжен, что я намереваюсь предложить пример поинтереснее: мы напишем приложение, рассказывающее сказку. Основа приложения – два класса. Каждый класс содержит члены двух видов: атрибуты и методы. Атрибуты – это данные, принадлежащие самому классу. Совокупное значение атрибутов определяет состояние объекта или класса. Метод – это последовательность операторов, изменяющих значения атрибутов и, как следствие, состояние класса или объекта.

В каждой интересной сказке должен быть свой герой. Предлагаю взглянуть на класс, который реализует нашего героя. Итак, по порядку. Следующий код необходимо сохранить в файл **FirstHero.java**.

```
public class FirstHero {
  private String quest;
  public FirstHero(String Quest) {
    this.quest = Quest;
  }
  public void setQuest(String Quest){
    this.quest = Quest;
  };
  public String getQuest(){
    return quest;
  };
}
```

Расположение одного класса в одном файле считается хорошим тоном среди программистов на Java – так что приучайтесь к нему сразу. Имя файла должно совпадать с названием класса, расширение у файла – **.java**.

Первый созданный нами класс содержит один защищенный атрибут (private – модификатор, зарезервированное слово, которое обеспечивает возможность обращения к члену класса только внутри класса) и три открытых метода (public – модификатор, зарезервированное слово, которое обеспечивает возможность обращения к члену класса из любого другого класса.). Хорошо спроектированный класс не имеет открытых атрибутов, что позволяет быть уверенным в правильной работе с ними. Доступ к защищенным атрибутам осуществляется через открытые методы. Такая организация класса позволяет разработчику через методы реализовать различные проверки и исключить неправильное поведение объекта. Класс представляет собой объединение атрибутов и методов в единое целое – подобное объединение называется инкапсуляцией.

Вы можете заметить, что первый метод (третья строка представленного ранее листинга) имеет то же название, что и класс – FirstHero. Такой метод называется конструктором. Конструктор – это метод, который используется при создании экземпляра класса для задания внутреннего состояния объекта, например:

```
FirstHero ourNewHero = new FirstHero("Спасти царевну");
```

Единственный атрибут нашего класса quest имеет тип строка (String). Для установки значения атрибута quest используется метод setQuest. Метод setQuest является public и не возвращает никакого значения. Для обозначения методов, не возвращающих значение, используется зарезервированное слово void. Для вызова метода setQuest необходим обязательный параметр: Quest типа String. Для получения значения атрибута quest используется метод getQuest, который вызы- »

вается без параметров, является открытым и возвращает значение типа String.

Вызов описанных выше методов для объекта ourNewHero осуществляется следующим образом:

```
ourNewHero.getQuest();
ourNewHero.setQuest("какая-то строка");
```

Парные методы, различающиеся приставками set и get, принято создавать для работы с закрытыми атрибутами. Это обязательное условие для реализации объекта в виде JavaBean. Что такое JavaBean, мы обсудим позже – сейчас запомните, что такое наименование методов позволяет избавиться от ряда проблем в больших системах.

Приступим к написанию самой сказки. Возьмите следующий код и поместите его в файл **Story.java**.

```
public class Story {
    public static void main(String[] args) {
        FirstHero ourNewHero = new FirstHero("Спасти царевну ");
        System.out.println("Наш герой хочет отправиться в путь и "
                + ourNewHero.getQuest());
    }
}
```

Это класс Story, он имеет всего один метод main, который является статичным (static-методы и переменные принято называть переменными класса и методами класса, т.к. они общие для всех объектов) и используется для работы с классом в целом. Метод main – это стартовая точка для запуска программы, с которой начинается интерпретация кода.

В ходе интерпретации метода main создается новый объект ourNewHero, а также вызывается метод println объекта out, который обеспечивает вывод в стандартный выходной поток строковой переменной.

Откомпилируем код и увидим, что у нас получилось. Для компиляции переместитесь в каталог, где вы разместили свои файлы и выполните следующую команду:

```
# javac *.java
```

При желании вы можете откомпилировать классы по отдельности, следующим образом:

```
# javac Story.java
# javac FirstHero.java
```

В итоге вы получите два файла **Story.class** и **FirstHero.class**. Теперь посмотрим на результат:

```
# java Story
Наш герой хочет отправиться в квест и Спасти царевну
```

Первая программа на Java готова и работает! Вы создали два класса и научились выводить информацию в консоль. Но мы забыли о комментариях...

Комментарии бывают нескольких типов:

- Строчные начинаются с символа **//** и длятся до окончания строки.
- Многострочные комментарии заключаются между **/*** и ***/**.
- Комментарии JavaDoc – это многострочные комментарии, заключенные между **/**** и ***/**, например:

```
/**
 * Класс является прототипом объекта "герой"
 * @author Chernousov Anton
 * @version 0.1
 */
public class FirstHero {
...
/**
    * Метод предназначен для установки внутренней переменной quest
    * в значение Quest.
    * У этого метода один параметр
    * @param Quest это переменная означает задание для Героя
    */
    public void setQuest(String Quest)
...
```

Комментарии JavaDoc предназначены для формирования документации к проекту из исходных кодов. Если добавить в файл **FirstHero.java** только что приведенные строки, то с помощью команды **javadoc -author -version *.java** можно сформировать документацию к нашему проекту.

JavaDoc позволяет создавать подробную документацию, которая просто необходима для любого достаточно крупного проекта.

Наследование, интерфейсы их реализация

Наша сказка написана, но что делать, если проект большой и над ним работает много людей? Как реализовать одновременное программирование разных классов так, чтобы результаты, полученные отдельными разработчиками, можно было без труда соединить в единое целое? Для решения этой задачи, а также задачи множественного наследования в Java существуют интерфейсы – interface. Но обо всем по порядку.

Наследование – это одна из парадигм программирования, которая связана с инкапсуляцией. Суть наследования заключается в следующем: если есть класс, инкапсулирующий в себе некоторые атрибуты и методы, то класс, наследующий его, автоматически наследует все его атрибуты и методы.

```
public class SecondHero extends FirstHero{
    public SecondHero(String Quest) {
        super(Quest + " Несмеяну");
    }
}
```

Расположенный выше код объявляет класс SecondHero, который с помощью зарезервированного слова extends наследует (расширяет) класс FirstHero. В Java класс может наследовать (расширять) только один класс. При наследовании класса FirstHero, содержащего конструктор, обязательно требуется переопределить его (например, как это делается в методе SecondHero). При выполнении оператора super(Quest) будет вызван аналогичный метод из суперкласса (класс-родителя). Строкой super(Quest + "Несмеяну") мы переопределяем поведение конструктора. Естественно, класс SecondHero может содержать новые атрибуты и методы.

Для проверки механизма наследования сохраните приведенный текст класса SecondHero в файл **SecondHero.java** и в файле **Story.java** измените имя класса FirstHero на SecondHero. Откомпилируйте и запустите программу, в результате вы должны увидеть следующее:

```
# java Story
Наш герой хочет отправиться в путь и Спасти царевну Несмеяну
```

При разработке программы, перед тем, как приступить к программированию, сначала проводится процесс проектирования (например, через UML-нотации). В результате проектирования появляется документация к проекту, а также спецификации классов. Интерфейс – это явно указанная спецификация набора методов, которые должны быть представлены в классе, реализующий эту спецификацию. В серьезном проекте на момент начала программирования перед программистом оказываются интерфейсы, которые он должен реализовать в классах.

Интерфейс в Java – это специальный класс, в котором отсутствует реализация! Давайте напишем интерфейс для нашего класса FirstHero. Следует скопировать следующий далее код и поместить его в файл **Hero.java** – именно этот интерфейс мы будет использован далее. Как вы видите, в отличие от других классов, интерфейс объявляется с помощью зарезервированного слова interface. В нашем случае интерфейс представляет собой следующий набор методов:

```
public interface Hero {
    public void setQuest(String Quest);
    public String getQuest();
    }
```

Чтобы класс FirstHero реализовывал интерфейс Hero, необходимо изменить первую строчку в файле **FirstHero.java** на public class FirstHero implement Hero {. Интерфейс может наследовать (расширять) множество других интерфейсов. Наример:

```
public interface Lord {
    public String sendToQuest();
```

```
    public void setBodyguard(Hero hisHero);
    public Hero getBodyguard();
}

public interface King extends Hero, Lord{
    public void setWife (String PrincesName);
    public String getWife();
}
```

Класс может одновременно наследовать класс и реализовывать один или несколько интерфейсов. Например, как в классе LandLord, который наследует класс FirstHero и реализует интерфейс Lord:

```
public class LandLord extends FirstHero implements Lord{
    private Hero bodyguard;
    public LandLord(String Quest, Hero hisHero) {
        super(Quest);
        this.bodyguard = hisHero;
    }
    public String sendToQuest() {
        bodyguard.setQuest(this.getQuest());
        return null;
    }
    public void setBodyguard(Hero hisHero) {
        this.bodyguard = hisHero;
    }
    public Hero getBodyguard() {
        return bodyguard;
    }
}
```

Чтобы окончательно закрыть тему наследования, необходимо поговорить про абстрактные классы – те, что занимают промежуточное место между интерфейсами и реальными классами. Объявление абстрактного класса или метода выполняется с помощью специального слова abstract:

```
abstract public class Knight
```

Абстрактный класс представляет собой неполную реализацию всех объявленных методов, и его можно наследовать, как и любой из классов. Обязательным условием при наследовании является полная реализация абстрактных методов.

Полиморфизм

Полиморфизм – это концепция, реализованная в Java, которая позволяет одному и тому же объекту выступать в разных формах. Например, экземпляр класс LandLord можно использовать как объект класса FirstHero. Это стало возможным благодаря тому, что класс LandLord наследует класса FirstHero.

Перегруженные методы

Иногда при создании сложных (и не очень) систем есть необходимость реализовать два различных метода, названых одинаково – это возможно с помощью перегруженных (overloading) методов. Выбор реализации зависит от типа обьектов, переданных в качестве параметров.

```
public class GreatKing extends LandLord implements King{
    private String wife;
    private boolean theftedWife;

    public GreatKing(String Quest, Hero hisHero) {
        super(Quest, hisHero);
    }

    public void setWife(String PrincesName) {
        this.wife = PrincesName;
        this.theftedWife = false;
    }

    public String getWife() {
        if (theftedWife){return null;}
        else{ return wife;}
    }

    public String getWife(boolean flag) {
        if (flag){ this.theftedWife = true;}
        return wife;
    }
}
```

В представленном классе GreatKing существуют две реализации перезагруженного метода getWife, которые можно вызвать следующим образом:

```
    GreatKing ourKing = new GreatKing("Править мудро",
(Hero) new FirstHero("Охранять господина"));
    String hisWife = ourKing.getWife();
    String theftedHisWife = ourKing.getWife(true);
```

Вместо заключения

Ну вот, мы создали всех героев. Разместите интерфейсы Lord, King, а также классы LandLord и GreatKing в соответствующих файлах. Немного измените класс Story, чтобы завершить сказку:

```
public class Story {
    public static void main(String[] args) {
        // создадим наших героев
        Hero ourNewHero = (Hero) new FirstHero("Спасити царевну");
        GreatKing ourKing = new GreatKing("Править мудро",
(Hero) new FirstHero("Охранять господина"));
        LandLord ourNewLandLord = new LandLord("Похитить царевну",
(Hero) new FirstHero("Охранять господина"));
        // Женим нашего короля
        ourKing.setWife("Несмеяна");
        // И начнем повествование
        System.out.println("Жил был мудрый король, и должен был он "
+ ourKing.getQuest());
        System.out.println("Была у него жена, и звали ее " + ourKing.
getWife());
        System.out.println("Был в королевстве коварный Визирь и хотел
он "
+ ourNewLandLord.getQuest());
        System.out.println("И сделал это");
        String wife = ourKing.getWife(true);
        //
        System.out.println("А наш герой жаждал " + ourNewHero.
getQuest()
+ " ..и добился этого");
        ourKing.setWife(wife);
        System.out.println("Король был счастлив.");
        System.out.println("Ну а что случилось дальше вы можете
придумать сами...");
    }
```

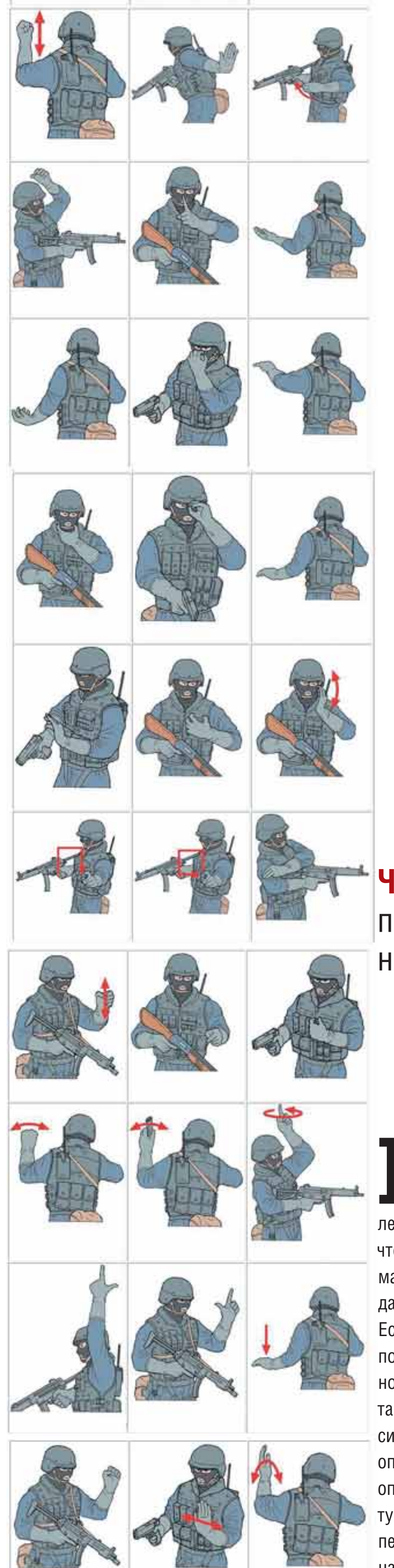
Откомпилируйте и насладитесь результатом выполнения урока.

В следующей статье мы поговорим о простых числах и других типах, вычислениях, циклах, массивах и списках. LXF

» Через месяц Мы изучим типы данных Java, циклы, массивы и списки.







Сигналы

We know when to kiss
And we know when to kill
If we can`t have it all
Then nobody will

Garbage. The World is not Enough

ЧАСТЬ 4: Сигналы традиционно считаются трудными в использовании, но при всем этом они незаменимы, когда речь заходит о настоящих приложениях Unix. Разберитесь в них вместе с **Андреем Боровским**!

Главное отличие сигналов от других средств взаимодействия между процессами заключается в том, что их обработка программой обычно происходит сразу же после поступления сигнала (или не происходит вообще), независимо от того, что программа делает в данный момент. Сигнал прерывает нормальный порядок выполнения инструкций в программе и передает управление специальной функции – обработчику сигнала. Если обработка сигнала не приводит к завершению процесса, то по выходе из функции-обработчика выполнение процесса возобновляется с той точки, в которой оно было прервано. У программ также есть возможность приостановить обработку поступающих сигналов временно, на период выполнения какой-либо важной операции. В традиционной терминологии приостановка получения определенных сигналов называется блокированием. Если для поступившего сигнала было установлено блокирование, сигнал будет передан программе, как только она разблокирует данный тип сигналов. Этим блокирование отличается от игнорирования сигнала, при котором сигналы соответствующего типа никогда не передаются программе. Следует помнить, что не все сигналы могут быть проигнорированы. Например, при получении программой сигнала принудительного завершения SIGKILL система ничего не сообщает программе, а просто прекращает ее работу. Таким образом, преимущество сигналов перед другими средствами межпроцессного взаимодействия заключается в том, что посылать программе сигналы можно в любой момент ее работы, не дожидаясь наступления каких-то особых условий. Источником сигналов может быть как сам операционная система, так и другие пользовательские программы. Если вам показалось, что сигналы похожи на прерывания, то вы совершенно правы. Для реализации сигналов действительно используются программные прерывания.

Нужно ли обрабатывать сигналы в вашей программе? Большинство программ не делают этого. В случае программирования для графических оболочек многие функции сигналов берут на себя механизмы сообщений графической оболочки. Тем не менее, есть целый ряд программ (например, демоны и консольные многопоточные приложения), в которых обработка сигналов необходима.



Большинству сигналов системы присвоена конкретная роль и, хотя у программиста существует возможность использовать сигналы для передачи произвольной информации, не соответствующей их стандартному назначению, делать этого не рекомендуется. Собственно говоря, с помощью сигналов можно передать не так уж и много информации – только номер сигнала (хотя на платформе x86, например, можно было бы организовать и передачу дополнительных параметров). Скудость данных, передаваемых сигналами, не удивительна, если учесть, что по умолчанию большинство сигналов просто завершают работу программы. При этом в некоторых случаях на диске сохраняется образ памяти выгруженной программы (знаменитый core dump). Соответственно и программа-источник сигнала обычно не ждет никакого ответа от программы-приемника. Номерам сигналов соответствуют константы, определенные в файле **signal.h**. Имена всех этих констант начинаются с префикса SIG, за которыми следует сокращенное название сигнала. Стандарт POSIX определяет две группы сигналов – «классические» сигналы Unix и сигналы реального времени. В отличие от классических сигналов сигналы реального времени всегда буферизуются, так что программа получит все посланные ей сигналы. В этой статье мы рассмотрим только классические сигналы Unix, каковых в Linux насчитывается 31. Этим сигналам назначены номера с 1 до 31 (номер 0, так называемый null-сигнал, имеет особый смысл). Полный список сигналов можно получить из заголовочного файла **signal.h**. Мы же рассмотрим несколько наиболее интересных сигналов.

» **Сигнал SIGHUP** (номер 1) изначально был предназначен для того, чтобы информировать программу о потере связи с управляющим терминалом (терминалы часто подключались к системе с помощью модемов, так что название сигнала происходит от hung up – повесить трубку). Кроме того, сигнал SIGHUP посылается приложению в том случае, если процесс-лидер сессии завершил свою работу. Многие программы-демоны, у которых нет лидера сессии, также обрабатывают этот сигнал. В ответ на получение SIGHUP демон обычно перезапускается (или просто повторно читает файл конфигурации). По умолчанию программа, получившая этот сигнал, завершается.

» **Сигнал SIGINT** (номер 2) обычно посылается процессу, если пользователь терминала дал команду прервать процесс (обычно эта команда – сочетание клавиш Ctrl-C) .

» **Сигнал SIGABRT** (номер 6) посылается программе в результате вызова функции abort(3). В результате программа завершается с сохранением на диске образа памяти.

» **Сигнал SIGKILL** (номер 9) завершает работу программы. Программа не может ни обработать, ни игнорировать этот сигнал.

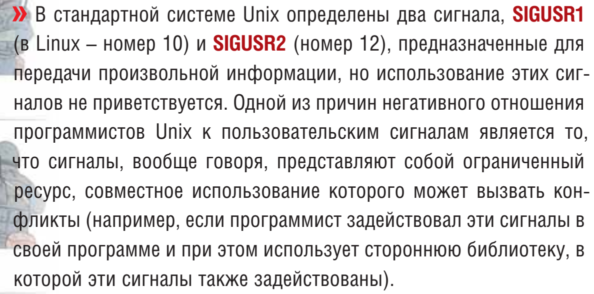
» **Сигнал SIGSEGV** (номер 11) посылается процессу, который пытается обратиться к не принадлежащей ему области памяти. Если обработчик сигнала не установлен, программа завершается с сохранением на диске образа памяти.

» **Сигнал SIGTERM** (номер 15) вызывает «вежливое» завершение программы. Получив этот сигнал, программа может выполнить необходимые перед завершением операции (например, высвободить занятые ресурсы). Получение SIGTERM свидетельствует не об ошибке в программе, а о желании ОС или пользователя завершить ее.

» **Сигнал SIGCHLD** (номер 17) посылается процессу в том случае, если его дочерний процесс завершился или был приостановлен. Родительский процесс также получит этот сигнал, если он установил режим отслеживания сигналов дочернего процесса и дочерний процесс получил какой-либо сигнал. По умолчанию сигнал SIGCHLD игнорируется.

» **Сигнал SIGCONT** (номер 18) возобновляет выполнение процесса, остановленного сигналом SIGSTOP.

» **Сигнал SIGSTOP** (номер 19) приостанавливает выполнение процесса. Как и SIGKILL, этот сигнал не возможно перехватить или игнорировать.



» **Сигнал SIGTSTP** (номер 20) приостанавливает процесс по команде пользователя (обычно эта команда – сочетание клавиш Ctrl-Z).

» **Сигнал SIGIO/SIGPOLL** (в Linux обе константы обозначают один сигнал – номер 29) сообщает процессу, что на одном из дескрипторов, открытых асинхронно, появились данные. По умолчанию этот сигнал, как ни странно, завершает работу программы.

» В стандартной системе Unix определены два сигнала, **SIGUSR1** (в Linux – номер 10) и **SIGUSR2** (номер 12), предназначенные для передачи произвольной информации, но использование этих сигналов не приветствуется. Одной из причин негативного отношения программистов Unix к пользовательским сигналам является то, что сигналы, вообще говоря, представляют собой ограниченный ресурс, совместное использование которого может вызвать конфликты (например, если программист задействовал эти сигналы в своей программе и при этом использует стороннюю библиотеку, в которой эти сигналы также задействованы).

Если вы не знали, то вам, возможно, будет интересно узнать, что обработка сигналов является частью стандарта языка Си и, как таковая, поддерживается даже на платформе Microsoft Windows. Однако, стандартный интерфейс сигналов Си, основанный на функции signal(), довольно неуклюж (недостатки интерфейса сигналов Си подробно описаны в книге [2]), так что мы воспользуемся более совершенным вариантом интерфейса, основанным на функции sigaction(2). Для демонстрации работы обработки сигналов мы напишем небольшую программу (файл **sigdemo.c** на компакт-диске).

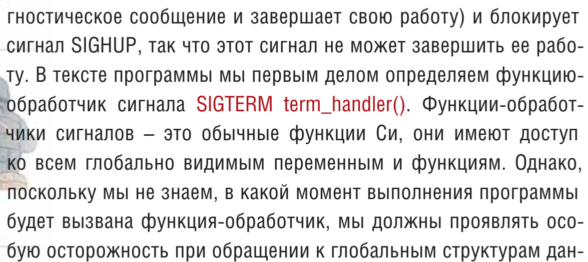
```
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>

void term_handler(int i) {
  printf ("Terminating\n");
  exit(EXIT_SUCCESS);
}

int main(int argc, char ** argv) {
  struct sigaction sa;
  sigset_t newset;
  sigemptyset(&newset);
  sigaddset(&newset, SIGHUP);
  sigprocmask(SIG_BLOCK, &newset, 0);
  sa.sa_handler = term_handler;
  sigaction(SIGTERM, &sa, 0);
  printf("My pid is %i\n", getpid());
  printf("Waiting...\n");
  while(1) sleep(1);
  return EXIT_FAILURE;
}
```









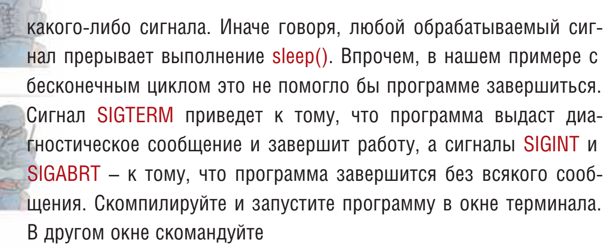


Наша программа делает две вещи: обрабатывает сигнал SIGTERM (при получении этого сигнала программа выводит диагностическое сообщение и завершает свою работу) и блокирует сигнал SIGHUP, так что этот сигнал не может завершить ее работу. В тексте программы мы первым делом определяем функцию-обработчик сигнала SIGTERM term_handler(). Функции-обработчики сигналов – это обычные функции Си, они имеют доступ ко всем глобально видимым переменным и функциям. Однако, поскольку мы не знаем, в какой момент выполнения программы будет вызвана функция-обработчик, мы должны проявлять особую осторожность при обращении к глобальным структурам данных из этой функции. Единственным параметром нашего варианта функции-обработчика сигнала (в Unix-системах существует и другой вариант) является переменная типа int, в которой передается номер сигнала, вызвавшего обработчик. Нам этот номер не нужен, поскольку мы знаем, что только один сигнал – SIGTERM, может вызвать нашу функцию, однако, в принципе, ничто не мешает нам использовать одну функцию для обработки нескольк- »









» ких разных сигналов, и тогда параметр функции-обработчика будет иметь для нас смысл. Функция-обработчик не возвращает никакого значения, что вполне логично, так как она вызывается не нашей программой, а неким системным компонентом. Особый интерес представляет завершение программы из обработчика сигнала. Назначение обработчика сигналу SIGTERM означает, что «умолчательное» действие сигнала – завершение программы не будет выполняться автоматически, и нам необходимо (если, конечно, мы хотим, чтобы этот сигнал завершал программу) позаботиться обо всем явным образом. Если вы закомментируете вызов exit() в нашем примере, то увидите, что программа не будет завершаться по получении сигнала SIGTERM. В принципе, вы можете придать сигналу SIGTERM совершенно иной смысл, например, оповещать программу о наступлении времени вашей любимой телепередачи (или о выходе нового номера журнала Linux Format), однако назначать стандартным сигналам нестандартные действия категорически не рекомендуется. Обработчик SIGTERM предназначен для того, чтобы, по требованию системы или пользователя, программа могла быстро и элегантно закончить текущую задачу и завершить свое выполнение. Именно этим обработчик и должен заниматься.

Перейдем теперь к тексту главной функции программы. Установка и удаление обработчиков сигналов осуществляются функцией sigaction(2). Первым параметром этой функции является номер сигнала, а в качестве второго и третьего параметров следует передать указатели на структуру sigaction. Эта структура содержит данные об операции, выполняемой над обработчиком сигнала. Второй параметр sigaction() служит для передачи новых значений для обработки сигнала, а третий – возвращает ранее установленные значения. В таблице 1 приводится краткое описание полей структуры sigaction.

Поля структуры sigaction

Поле	Значение
sa_handler	Указатель на функцию обработчик сигнала или константа
sa_mask	Маска сигналов, блокируемых на время вызова обработчика
sa_flags	Дополнительные флаги

Поле sa_handler должно содержать либо адрес функции-обработчика, либо специальную константу, указывающую, что нужно делать с сигналом. Константа SIG_IGN указывает, что сигнал следует игнорировать, а константа SIG_DFL – что нужно восстановить обработку сигнала, заданную системой по умолчанию. Поле sa_mask позволяет заблокировать некоторое множество сигналов на время выполнения обработчика данного сигнала. Делается это для того, чтобы обработка других сигналов не могла прервать обработку данного (это может быть необходимо, особенно, если один обработчик обрабатывает несколько разных сигналов). Параметр sa_flags позволяет задать ряд флагов для выполнения более тонкой настройки обработчика сигналов. Например, флаг SA_RESETHAND указывает, что после завершения обработки сигнала заданным обработчиком должен быть восстановлен обработчик, заданный по умолчанию, так что все последующие сигналы будут обрабатываться «умолчательным» обработчиком.



В результате вызова функции sigaction() мы устанавливаем обработчик сигнала SIGTERM. Затем наша программа распечатывает значение PID (это значение понадобится нам для вызова команды kill) и входит в бесконечный цикл, из которого она может быть выведена одним из сигналов. Следует отметить, что функция sleep() прерывается (возобновляет выполнение программы раньше срока) если возвращает управление обработчик какого-либо сигнала. Иначе говоря, любой обрабатываемый сигнал прерывает выполнение sleep(). Впрочем, в нашем примере с бесконечным циклом это не помогло бы программе завершиться. Сигнал SIGTERM приведет к тому, что программа выдаст диагностическое сообщение и завершит работу, а сигналы SIGINT и SIGABRT – к тому, что программа завершится без всякого сообщения. Скомпилируйте и запустите программу в окне терминала. В другом окне скомандуйте

```
kill <PID>
```

где PID – идентификатор процесса программы. Вы увидите, что перед тем как завершиться программа выдает диагностическое сообщение, тогда как при завершении с помощью Ctrl-C никакого сообщения не выводится.

Рассмотрим теперь блокировку сигналов. Поскольку игнорирование сигнала устанавливается функцией sigaction(), можно было бы ожидать, что и блокировка устанавливается этой же функцией, но это не так. Поскольку нам, как правило, приходится блокировать несколько сигналов сразу, для блокировки существует специальная функция sigprocmask(2), которая оперирует наборами сигналов (signal sets). Разделение интерфейса между несколькими функциями вызвано еще и требованиями многопоточности. Параметры, устанавливаемые sigaction(), действительны для всей программы в целом, тогда как блокировку сигналов потоки осуществляют независимо друг от друга. Наборы сигналов хранятся в переменных специального типа – sigset_t, а операции над ними осуществляются с помощью специальных функций. Функция sigemptyset() инициализирует набор сигналов пустыми значениями, а функция sigfillset() устанавливает все возможные значения в наборе. Используемая нами функция sigaddset() добавляет значение сигнала в набор, а функция sigdelset() удаляет сигнал из набора. После того как набор сигналов сформирован, мы передаем его функции sigprocmask(), которая выполняет блокировку и разблокировку сигналов. Первым параметром этой функции должна быть одна из констант, определяющих операцию над заданными сигналами. Константа SIG_BLOCK указывает, что сигналы из нового набора должны быть добавлены к списку уже заблокированных сигналов. Константа SIG_SETMASK указывает, что новый набор блокируемых сигналов должен заменить уже существующий (при этом заблокированные ранее сигналы будут разблокированы, если они не заблокированы в новом наборе), а константа SIG_UNBLOCK указывает на необходимость разблокировать сигналы, переданные в наборе. В нашей программе мы блокируем сигнал SIGHUP и вы можете видеть, что программа не обрабатывает этот сигнал. Послать нашей программе сигнал SIGHUP вы можете с помощью консольной команды

```
kill –s 1 <PID>
```

где PID – идентификатор процесса.

Сигналы прерывают нормальный порядок выполнения программы и могут завершить работу программы, не способной завершиться иным образом. Но иногда бывает так, что программе просто нечего делать до тех пор, пока она не получит какой-либо сигнал. Иначе говоря, программу нужно заставить ждать появления сигнала, по возможности не нагружая процессор. Такая ситуация может возникнуть, например, в многопоточном приложении, когда нужно синхронизировать завершение нескольких потоков. Ожидание сигнала можно реализовать с помощью цикла, проверяющего значение флажка, который может сбросить обработчик сигнала. В некоторых случаях (таких как рассмотренный выше пример) можно реализовать ожидание и с помощью бесконечного цикла. Очевидно, однако, что эти методы не эффективны и не элегантны. В POSIX-системах существует специальная функция sigwait(3), которая «усыпляет» процесс до тех пор, пока процессу не будет передан один из заданного набора сигналов. Модифицируем нашу программу так, чтобы вместо бесконечного цикла она входила в цикл ожидания сигнала SIGHUP (файл **swdemo.c** на компакт-диске):






```
sigprocmask(SIG_BLOCK, &newset, 0);
while(!sigwait(&newset, &sig))
  printf("SIGHUP recieved\n");
```

Первым параметром функции sigwait() является указатель на набор сигналов, получения которых будет ждать функция. Во втором параметре sigwait() вернет номер того сигнала, который возобновил работу программы (эта информация может быть полезна, если установлено несколько ожидаемых сигналов). Перед тем как вызывать sigwait(), набор ожидаемых сигналов следует заблокировать с помощью функции sigprocmask(), иначе, при получении сигнала, вместо выхода из sigwait() будет вызван соответствующий обработчик. Сигнал, который возобновил работу программы после вызова sigwait(), уже не может быть перехвачен назначенным ему обработчиком. В нашем примере мы «усыпляем» программу до тех пор, пока она не получит сигнал SIGHUP, распечатываем соответствующее сообщение и снова усыпляем (функция sigwait() возвращает 0, если ее вызов прошел успешно). В то время, когда программа приостановлена в ожидании некоторых сигналов, обработчики всех не заблокированных и не игнорируемых сигналов выполняются обычным образом.

Функцию sigwait() можно использовать и для исследования сигналов. На компакт-диске вы найдете программку **siglog.c**, которая распечатывает информацию о каждом поступившем сигнале (естественно, исследуются только те сигналы, которые могут быть заблокированы). Рассмотрим здесь фрагмент этой программы:

```
sigset_t sset;
int sig;
...
sigfillset(&sset);
sigdelset(&sset, SIGTERM);
sigprocmask(SIG_SETMASK, &sset, 0);
while(!sigwait(&sset, &sig))
printf("Signal %i - %s\n", sig, sys_siglist[sig]);
```

С помощью вызовов sigfillset() и sigdelset() мы создаем набор из всех сигналов, за исключением сигнала SIGTERM (этот сигнал понадобится нам для того, чтобы мы могли завершить работу программы). Далее мы блокируем сигналы набора sset и вызываем для них функцию sigwait(). Функция вернет управление при получении любого сигнала, кроме SIGTERM (для которого назначен отдельный обработчик). Получив новый сигнал, мы распечатываем информацию о нем. Массив char * sys_siglist[] определен в стандартной библиотеке *glibc*. Этот массив содержит наименования сигналов на «человеческом» языке (эти наименования можно использовать при выводе диагностических и отладочных сообщений). Наименования расположены так, чтобы их индексы в массиве соответствовали номерам сигналов. Те же данные возвращает и функция strsignal(), единственным параметром которой является номер сигнала.

На протяжении всей этой статьи мы занимались обработкой сигналов, но не их генерацией. Поскольку основным источником сигналов является операционная система, нам и в «реальной жизни» чаще приходится заниматься именно обработкой. Однако, в заключение статьи следует рассмотреть и функции генерации сигналов. Для генерации сигналов в Unix предусмотрены две функции – kill(2) и raise(3). Первая функция предназначена для передачи сигналов любым процессам, к которым владелец данного процесса имеет доступ, а с помощью второй функции процесс может передать сигнал самому себе. Как это обычно принято в мире Unix, семантика вызова функции kill() совпадает с семантикой одноименной команды ОС. У функции kill() два аргумента – PID процесса-приемника и номер передаваемого сигнала. С помощью функции kill() как и с помощью одноименной команды можно передавать сообщения не только конкретному процессу, но и группе процессов. Таблица 2 демонстрирует поведение функции kill() в зависимости от значения PID:

Поля структуры sigaction

Поле	Значение
PID > 1	Сигнал посылается процессу с соответствующим PID.
PID == 0	Сигнал посылается всем процессам из той же группы, что и процесс-источник.
PID < 0	Сигнал посылается всем процессам, чей идентификатор группы равен абсолютному значению PID.
PID == 1	Сигнал посылается всем процессам системы.

Вызов

```
raise(sig);
```

эквивалентен вызову

```
kill(getpid(), sig);
```

Так же как и для других примитивов IPC, для сигналов действует система прав доступа, основанная на правах доступа владельцев процессов. Процесс-приемник получит сигнал только в том случае, если у процесса-источника есть соответствующие права. С помощью функции kill() можно проверить, существует ли в системе процесс с заданным PID, не посылая процессу никаких сигналов. Для этого предназначен псевдо-сигнал с номером 0. Если соответствующего процесса не существует, функция kill() вернет значение 1, соответствующее об ошибке. В любом случае, сигнал не будет отправлен.

Читателей, полюбивших обработку сигналов, я могу обрадовать тем, что мы рассмотрели далеко не все функции, связанные с сигналами. При изучении документации вас ждет еще много полезного и приятного, мы же закончим на этом наше знакомство с сигналами.

Мы можем столкнуться с необходимостью обработки сигналов при программировании, например, процессов-демонов. Ими мы и займемся в следующей статье. LXF

Литература:

1. D. P. Bovet, M. Cesati, **Understanding the Linux Kernel, 3rd Edition, O'Reilly**, 2005
2. W. R. Stevens, S. A. Rago, **Advanced Programming in the UNIX® Environment: Second Edition**, Addison Wesley Professional, 2005

Через месяц Мы породим монстра... то есть демона.

























Алгебра и начала

ЧАСТЬ 4: Maxima способна прорешать половину Демидовича за десять минут. Не верите? Спросите у **Тихона Тарнавского**!

Из встроенного функционала *Maxima* в первую очередь стоит обратить внимание на несколько групп функций: работу с пределами, дифференцирование, интегрирование, поиск решений уравнений – как «просто», так и дифференциальных.

«Предельничаем»...

Собственно полноценных функций для нахождения предела существует в *Maxima* аж одна. Но зато какая! Она может принимать три различных варианта списка аргументов, и кроме того, на ее действие влияют еще и три флага. Но давайте по порядку. Зовут эту функции вполне соответственно ее действию: limit; и в самом стандартном варианте ее вызов выглядит как limit(выражение, переменная, точка), то есть то, что в математической записи выглядит как $\lim_{x\to a} f(x)$, в контексте *Maxima* запишется как limit(f(x), x, a):

(%i1) $\mathrm{limit}\left(\frac{x^2-1}{2x^2-x-1}, x, 1\right)$

(%o1) $\frac{2}{3}$

(%i2) $\mathrm{limit}\left(\frac{(1+m\,x)^n-(1+n\,x)^m}{x^2}, x, 0\right)$

(%o2) $-\frac{m\,(m-n)\,n}{2}$

(%i3) $\mathrm{limit}\left(\frac{(x^n-a^n)-n\,a^{n-1}\,(x-a)}{(x-a)^2}, x, a\right)$

(%o3) $\frac{n\,(a^n\,n-a^n)}{2\,a^2}$

Maxima может искать пределы не только в конечных точках, но и на бесконечности. Среди стандартных обозначений программы существуют универсальные названия для разных бесконечностей: плюс-бесконечность записывается через inf (от слова «infinity», как нетрудно догадаться), минус-бесконечность – через minf (от «minus infinity»); для комплексных чисел бесконечность, как известно, одна, и она (комплексная бесконечность) обозначается полным словом infinity. При работе с пределами все три обозначения могут как использоваться при вводе, так и возникать в виде найденного значения предела; отдельно здесь надо отметить один момент касательно работы с интерфейсом к *Maxima* в редакторе *TeXmacs*: символы inf и minf при выводе здесь отображаются в своей традиционной математической нотации, то есть как ∞ и $-\infty$; символ ∞ вместо inf можно, кроме того, использовать еще и при вводе.

(%i1) $\mathrm{limit}\left(\frac{\sqrt{x}+\sqrt[3]{x}+\sqrt[4]{x}}{\sqrt{2x+1}}, x, \infty\right)$

(%o1) $\frac{1}{\sqrt{2}}$

(%i2) $\mathrm{limit}\left(\sqrt{x+\sqrt{x+\sqrt{x}}}-x, x, \infty\right)$

(%o2) $-\infty$

Второй вариант вызова функции limit() – это расширенная версия первого: limit(выражение, переменная, точка, направление), для поиска односторонних пределов. Для предела справа в качестве «направления» указывается plus, для предела слева – minus:

(%i1) $\mathrm{limit}\left(\frac{\mathrm{abs}(\sin(x))}{x}, x, 0, \mathrm{plus}\right)$

(%o1) 1

(%i2) $\mathrm{limit}\left(\frac{\mathrm{abs}(\sin(x))}{x}, x, 0, \mathrm{minus}\right)$

(%o2) -1

» **Месяц назад** Мы занимались упрощением выражений.

Б.П.ДЕМИДОВИЧ

СБОРНИК ЗАДАЧ И УПРАЖНЕНИЙ ПО МАТЕМАТИЧЕСКОМУ АНАЛИЗУ

13-е издание, исправленное

Рекомендовано Государственным комитетом Российской Федерации по высшему образованию в качестве учебного пособия для студентов математических и физических специальностей высших учебных заведений

анализа

ОТДЕЛ VIII. КРАТНЫЕ И КРИВОЛИНЕЙНЫЕ ИНТЕГРАЛЫ

Пределы справа и слева еще иногда называют соответственно пределами сверху и снизу. Хотя правильнее в таком случае говорить полностью: «предел при **x**, стремящемся к **a** сверху», в том числе чтобы не создавать путаницы с верхним и нижним пределами, которые суть совершенно другое.

Кроме упомянутых выше бесконечностей, на выходе возможно появление и еще двух обозначений, на случай, если заданный предел не существует: ind (от слова indefinite – «неопределенный») и und (от слова undefined – опять же «неопределенный»). В документации первое из этих обозначений описано как «indefinite but bounded» (не определен, но ограничен), что дает предположить, что функция, не имеющая предела, при этом ограничена либо в окрестности предельной точки, либо на всей прямой. Какое из этих предположений имелось в виду, мне так и не удалось понять, потому как на практике ни одно из них не соответствует действительности. Вывода «ind» мне не удалось добиться ни на одной функции, радикально отличающейся от «канонической» (в том смысле, что фигурирующей в стандартном примере из комплекта) функции sin(1/x) [у нас аналогичный вывод получился и для (–1)^x – прим. ред.].

(%i1) $\mathrm{limit}\left(\sin\left(\frac{1}{x}\right), x, 0\right); \mathrm{limit}\left(\tan\left(\frac{1}{x}\right), x, 0\right)$

(%o1) ind
(%o2) und

Здесь все правильно, tan(1/x) не ограничена в окрестности нуля. А вот дальше начинаются чудеса:

(%i3) $\mathrm{limit}\left(a: \frac{1}{1-e^{\frac{1}{x}}}, x, 0\right)$

(%o3) und

(%i4) $\mathrm{limit}(a, x, 0, \mathrm{plus}); \mathrm{limit}(a, x, 0, \mathrm{minus}); \mathrm{limit}(a, x, \infty); \mathrm{limit}(a, x, \mathrm{minf})$

(%o4) 0
(%o5) 1
(%o6) $-\infty$
(%o7) ∞

(%i8) $\mathrm{limit}\left(a: \mathrm{atan}\left(e^{\frac{1}{x}}\right), x, 0\right)$

(%o8) und

(%i9) $\mathrm{limit}(a, x, 0, \mathrm{plus}); \mathrm{limit}(a, x, 0, \mathrm{minus}); \mathrm{limit}(a, x, \infty); \mathrm{limit}(a, x, \mathrm{minf})$

(%o9) $\frac{\pi}{2}$
(%o10) 0
(%o11) $\frac{\pi}{4}$
(%o12) $\frac{\pi}{4}$

Как видите, первая функция имеет конечные односторонние пределы в нуле, а вторая ограничена вообще на всей оси – и тем не менее... Но это, думаю, не столь критично: главное, что наличие любого из этих символов в качестве вывода дает нам понять, что искомого предела не существует.

Функция limit() в третьем варианте – limit(выражение) – предназначена уже не для поиска собственно пределов, а для упрощения выражений, содержащих символы inf и minf:

(%i1) $\mathrm{limit}\left(\frac{1}{\infty}\right)$

(%o1) 0

Выражения такого рода могут возникать, к примеру, при подстановках в формулы результатов вычисления каких-то других пределов или интегралов.

Такая способность – принимать различные списки аргументов – не является в *Maxima* чем-то особенным; она свойственна очень многим встроенным функциям, как и различное действие в зависимости от значений разнообразных переключателей. Это достаточно удобно: не нужно запоминать много разных имен функций (для поиска пределов, к примеру, используется исключительно функция limit); для вычисления производных, в том числе и частных, – функция diff (с которой мы уже бегло ознакомились в первой статье и сейчас продолжим это знакомство); для нахождения интегралов, как определенных, так и неопределенных – функция integrate (с которой мы тоже сегодня познакомимся). Имена наиболее часто используемых функций запомнить несложно, а о дополнительных ключах или флагах, в случае чего, можно прочитать во встроенной справке, набрав **? имя-функции**.

Об этих самых ключах к функции limit и осталось рассказать. Первый ключ называется lhospitallim и задает максимальное количество применений правила Лопиталя; название ключа и происходит от фамилии ученого, давшей название самому правилу, которая в оригинале пишется как L'Hospital. Напомню, правило это гласит, что в случае неопределенности вида 0/0 или ∞/∞ можно продифференцировать числитель и знаменатель – и предел от этого не изменится. Ограничитель количества применений этого правила нужен для того, чтобы избежать зацикливаний, которые могут случиться для бесконечно дифференцируемых функций, у которых в данной точке равны нулю либо бесконечности все производные. По умолчанию значение lhospitallim равно четырем, и мне не удалось сходу придумать пример, когда этого не хватает – ведь функция поиска предела использует не только правило Лопиталя, но и другие соотношения; и для всех задан- »

ных мною соотношений двух функций с корнями выше четвертого порядка в искомой точке предел был успешно найден и при умолчательном значении.

Второй ключ к функции limit – это флаг limsubst, который, будучи выставлен в true, позволяет этой функции производить подстановки внутрь неизвестных выражений. По умолчанию этот флаг равен false, что исключает ошибки вроде такой:

(%i1) $\mathrm{limit}\left(\frac{f(x)}{f(x+a)}, x, \infty\right), \mathrm{limsubst}: \mathrm{true}$

(%o1) 1

(%i2) $f(x) := \sin(x)\$''\%i1$

(%o3) und

И, наконец, последний дополнительный параметр – еще один флаг, по имени tlimswitch. По умолчанию он тоже выключен, а если его включить, функция limit будет, при невозможности найти предел другими способами, пытаться его найти путем разложения подпредельной функции в ряд Тейлора в окрестности заданной точки:

(%i1) $\mathrm{limit}\left(\frac{\sqrt{x+\sqrt{x+\sqrt{x}}}}{\sqrt{x+1}}, x, \mathrm{inf}\right)$

```
Quotient by a polynomial of higher degree
 -- an error.  Quitting.  To debug this try debugmode(true);
```

(%i2) $\mathrm{limit}\left(\frac{\sqrt{x+\sqrt{x+\sqrt{x}}}}{\sqrt{x+1}}, x, \mathrm{inf}\right), \mathrm{tlimswitch}: \mathrm{true}$

(%o2) 1

Но в случае поиска односторонних пределов, в тех точках, где они не равны между собой, то есть полного предела не существует, этим флагом нужно пользоваться с осторожностью: при его включении функция limit может вернуть в качестве полного предела один из односторонних:

(%i1) $\mathrm{limit}\left(\sqrt{\frac{1}{x}+\sqrt{\frac{1}{x}+\sqrt{\frac{1}{x}}}} - \sqrt{\frac{1}{x}-\sqrt{\frac{1}{x}+\sqrt{\frac{1}{x}}}}, x, 0\right), \mathrm{tlimswitch}: \mathrm{true}$

(%o1) 1

Реально у этой функции в точке ноль только предел справа равен единице; а предел слева – нулю.

И последнее: почему я употребил в начале по отношению к функции limit() слово «полноценная». Потому что кроме нее существует еще одна «недофункция» – tlimit(); она представляет собой фактически просто-напросто вызов самой функции limit() с поднятым флагом tlimswitch, то есть пытается при необходимости разложить «подпредельную» функцию в ряд Тейлора вне зависимости от реального значения этого флага. Другими словами вызов tlimit(аргументы) полностью аналогичен записи limit(аргументы), tlimswitch:true; только чуть короче. И аргументы она может принимать точно такие же.

...дифференцируем и интегрируем.

О функции diff я кое-что уже рассказывал в первой статье (*см. LXF81*), и здесь это «кое-что» только напомню. В двух упомянутых тогда вариантах вызова эта функция принимала один либо два аргумента. С двумя, diff(выражение, переменная), она возвращает производную от «выражения» по заданной переменной; с одним, diff(выражение) – полный дифференциал заданного выражения. Другими словами, запись diff(f, x) равнозначна математическому обозначению df/dx, а diff(f) – df.

Но это еще не все. Кроме одного либо двух, эта функция может также принимать любое нечетное число аргументов вида diff(выражение, переменная, порядок, переменная, порядок, ...) и возвращает при этом производную либо смешанную частную производную от выражения заданных порядков по заданным переменным. К примеру, diff(f, x, 3) означает d^3f/dx^3, а diff(f, x, 1, y, 2, z, 1) – $d^4f/dxdy^2dz$. Единственный флаг, имеющий прямое отношение к самой функции diff – это флаг derivabbrev, который влияет на отображение производных в ячейках вывода *Maxima*. По умолчанию он равен ***false***, и производные обозначаются в виде дробей с буквой d; если же его выставить в ***true***, производные будут отображаться в сокращенном виде, с переменными дифференцирования записанными в виде индексов:

(%i1) $\mathrm{diff}(f(x)^2, x, 4)$

(%o1) $2\,f(x)\left(\frac{d^4}{d\,x^4}\,f(x)\right)+8\left(\frac{d}{d\,x}\,f(x)\right)\left(\frac{d^3}{d\,x^3}\,f(x)\right)+6\left(\frac{d^2}{d\,x^2}\,f(x)\right)^2$

(%i2) $\mathrm{derivabbrev}: \mathrm{true}\$''\%i1$

(%o3) $2\,f(x)\,(f(x))_{x\,x\,x\,x}+8\,(f(x))_x\,(f(x))_{x\,x\,x}+6\,((f(x))_{x\,x})^2$

Кроме того, функция diff используется еще и для обозначения производных в дифференциальных уравнениях. Но об этом чуть позже, а сейчас перейдем к интегрированию.

Основная функция интегрирования называется, как я уже говорил, integrate и имеет два варианта вызова: для нахождения неопределенного и определенного интегралов. Первый выглядит как integrate(выражение, переменная), второй – как integrate(выражение, переменная, нижний-предел, верхний-предел):

(%i1) $\mathrm{integrate}\left(\sqrt{\frac{e^x-1}{e^x+1}}, x\right)$

(%o1) $\log\left(2\sqrt{e^{2x}-1}+2\,e^x\right)+\arcsin\left(e^{-x}\right)$

(%i2) $\mathrm{integrate}(\cos(x)\cos(2\,x)\cos(3\,x), x)$

(%o2) $\frac{\sin(6\,x)}{24}+\frac{\sin(4\,x)}{16}+\frac{\sin(2\,x)}{8}+\frac{x}{4}$

(%i3) $\mathrm{integrate}\left(\frac{x^{2n-1}}{x^n+1}, x\right)$

(%o3) $\frac{e^{n\log(x)}}{n}-\frac{\log\left(e^{n\log(x)}+1\right)}{n}$

(%i4) $\mathrm{integrate}(x^2\sqrt{a^2-x^2}, x, 0, a)$

Is a positive, negative, or zero?p

(%o4) $\frac{\pi\,a^4}{16}$

Вы, наверное, обратили внимание еще на один момент в ячейках %i4–%o4. Когда в выражении используется какой-либо независимый символ, результат, вообще говоря, может зависеть от значения этого символа. Если при этом о возможных значениях символа ничего не известно, то *Maxima* задаст вам один или несколько вопросов об этом значении, и решение будет искать в зависимости от ваших ответов на них. Так, в этом примере значение определенного интеграла напрямую зависит от знака параметра a:

(%i5) $\mathrm{integrate}(x^2\sqrt{a^2-x^2}, x, 0, a)$

Is a positive, negative, or zero?n

(%o5) $-\frac{\pi\,a^4}{16}$

Кроме обычных определенных интегралов *Maxima* умеет искать также и несобственные интегралы, то есть такие, у которых неограни-

чена либо область интегрирования, либо подынтегральная функция; и делается это все той же функцией integrate:

$$\text{(\%i1)}\quad \mathrm{integrate}\left(\frac{1}{1+e^{x}},x,1,\infty\right)$$

$$\text{(\%o1)}\quad \log\left(e+1\right)-1$$

$$\text{(\%i2)}\quad \mathrm{integrate}\left(\frac{1}{\sqrt{x}},x,0,1\right)$$

$$\text{(\%o2)}\quad 2$$

В случае, если искомый интеграл не сходится, будет выдано сообщение об ошибке, говорящее о том, что интеграл расходящийся:

$$\text{(\%i3)}\quad \mathrm{integrate}\left(\frac{1}{x^2},x,0,1\right)$$

```
Integral is divergent
 -- an error.  Quitting.  To debug this try debugmode(true);
```

В случае, если интеграл не может быть найден, он либо целиком возвращается в несовершенном виде, либо упрощается частично и на выходе получается некоторая формула, включающая в несовершенном виде интеграл той части подынтегрального выражения, которую проинтегрировать не удалось:

$$\text{(\%i1)}\quad \mathrm{integrate}\left(\frac{1}{x^4-4x^3+2x^2-7x-4},x\right)$$

$$\text{(\%o1)}\quad \frac{\log\left(x-4\right)}{73}-\frac{\int\frac{x^2+4x+18}{x^3+2x+1}dx}{73}$$

Кроме функций diff и integrate, в *Maxima* есть еще много разнообразных возможностей, связанных с производными и интегралами, в частности, функции для численного расчета значений определенных интегралов, а также инструменты, применимые при работе с дифференциальными и интегральными уравнениями. И разнообразны они настолько, что для того, чтобы рассмотреть подробно и с примерами их все, не хватило бы всего этого цикла. А более тезисно, хотя и на английском языке, они описаны в документации. Посему с этой темой на этом прекращаем и движемся дальше.

Решайте, сударь!

А дальше мы рассмотрим уравнения и их системы, и даже не столько рассмотрим, сколько порешим... то есть, конечно, порешаем. Уравнения и системы уравнений решаются в *Maxima* тоже одной и той же функцией, и тоже уже вам слегка знакомой по первой статье: это функция solve. Но прежде чем рассмотреть ее подробнее, нужно сказать пару слов о списках, или векторах, в *Maxima*; поскольку именно в виде списков solve возвращает корни, да и принимает параметры в случае решения системы уравнений, а не одного уравнения.

Синтаксис списков в *Maxima* весьма прост; это перечисление элементов в квадратных скобках: [элемент1, элемент2, ..., элементN]. Особенность – не в синтаксисе. Основное достоинство *Maxima*-льных списков в том, что их элементами могут быть совершенно любые выражения: символы, арифметические выражения, вызовы функций, присвоения, уравнения, другие списки... Додумать можете сами. Поэтому списки и во встроенных функциях применяются достаточно широко. Функция solve в своем простейшем варианте, для решения одиночного уравнения, в качестве аргументов никаких списков, напомню, не принимает (а принимает либо уравнение и символ, относительно которого его надо решать, либо только уравнение, если символ в нем всего один). А вот в качестве результата она уже и в таком варианте возвращает список, состоящий из всех корней заданного уравнения:

$$\text{(\%i1)}\quad \mathrm{eq}: \frac{x+1}{x^2+1}=\frac{x^2}{x+2}\$\ \mathrm{solve}(\mathrm{eq})$$

$$\text{(\%o2)}\quad \left[x=-\frac{\sqrt{5}-1}{2},x=\frac{\sqrt{5}+1}{2},x=-\frac{\sqrt{7}\,i+1}{2},x=\frac{\sqrt{7}\,i-1}{2}\right]$$

Как видите, функция solve находит все комплексные корни уравнения, а не только действительные.

К элементу списка можно обратиться с помощью тех же квадратных скобок, указав в них номер элемента после имени списка. Напомню, что равенство, переданное в качестве дополнительного параметра функции ev, означает подстановку переменной в вычисляемое выражение. Вот так мы можем осуществить проверку решения, подставив корень из выданного списка в исходное уравнение:

$$\text{(\%i3)}\quad \mathrm{eq},\%[1]$$

$$\text{(\%o3)}\quad \frac{1-\frac{\sqrt{5}-1}{2}}{\frac{\left(\sqrt{5}-1\right)^2}{4}+1}=\frac{\left(\sqrt{5}-1\right)^2}{4\left(2-\frac{\sqrt{5}-1}{2}\right)}$$

$$\text{(\%i4)}\quad \mathrm{ratsimp}(\%)$$

$$\text{(\%o4)}\quad \frac{\sqrt{5}-3}{\sqrt{5}-5}=\frac{\sqrt{5}-3}{\sqrt{5}-5}$$

Точно таким же образом можно обратиться и к любому другому элементу списка:

$$\text{(\%i5)}\quad \mathrm{eq},\%o2[2],\mathrm{ratsimp};\ \mathrm{eq},\%o2[3],\mathrm{ratsimp};\ \mathrm{eq},\%o2[4],\mathrm{ratsimp}$$

$$\text{(\%o5)}\quad \frac{\sqrt{5}+3}{\sqrt{5}+5}=\frac{\sqrt{5}+3}{\sqrt{5}+5}$$

$$\text{(\%o6)}\quad -1=-1$$

$$\text{(\%o7)}\quad -1=-1$$

Вообще говоря, в качестве первого аргумента функции solve можно задавать не только уравнение, а вообще любое выражение. При этом «корни выражения» (не являющегося уравнением) ищутся в том самом смысле, в каком эта фраза понимается в математике: корни выражения – это те значения переменной, на которых выражение обращается в ноль. Возможность такой записи позволяет, к примеру, легко найти критические точки любой непрерывной функции (а заодно и вычислить значения функции в этих точках):

$$\text{(\%i1)}\quad f: \frac{1+x-x^2}{1+x+x^2}\$$$

$$\text{(\%i2)}\quad \mathrm{solve}(\mathrm{diff}(f,x))$$

$$\text{(\%o2)}\quad [x=-2,x=0]$$

$$\text{(\%i3)}\quad f,\%[1];\ f,\%\mathrm{th}(2)[2]$$

$$\text{(\%o3)}\quad -\frac{5}{3}$$

$$\text{(\%o4)}\quad 1$$

В этом примере есть еще два важных момента. Первый – функция %th(). Она, как видно из контекста, вызывается как %th(n) и возвращает n-ю с конца ячейку вывода. Это, так же как и обозначения % и _, удобно, чтобы не обращать внимания на номера ячеек, и кроме того, применимо в командных файлах *Maxima*, которые могут загружаться в том числе и прямо из интерактивной сессии (с помощью функции load) – и тогда просто заранее неизвестно, начиная с какой ячейки данный файл загружен. И второй момент: здесь проиллюстрировано, что в *Maxima* операция индексирования списка доступна не только по отношению к именам переменных, но и к вызовам функций; другими словами, если функция возвращает список значений, мы можем выбрать одно конкретное из них, написав его номер в квадратных скобках прямо после вызова функции.

Вернемся к функции solve. А именно, перейдем теперь к решению систем уравнений. Для этого существует такой вариант записи: solve([уравнение1, уравнение2, ...], [переменная1, переменная2, ...]); либо сокращенный, аналогично варианту для одиночного уравнения: если количество уравнений и количество неизвестных равны, список неизвестных можно не писать: solve([уравнение1, уравнение2, ...]) (не забудьте квадратные скобки, иначе *Maxima* примет его за вариант с одним уравнением).

(%i1) $\mathrm{solve}([x^2+y^2=2, x+y=1])$

(%o1) $$\left[\left[y=-\frac{\sqrt{3}-1}{2}, x=\frac{\sqrt{3}+1}{2}\right], \left[y=\frac{\sqrt{3}+1}{2}, x=-\frac{\sqrt{3}-1}{2}\right]\right]$$

Здесь возвращается список из нескольких списков, каждый из которых соответствует одному решению системы. В качестве подстановок можно использовать как такие списки целиком (например, в данном контексте, %o1[1]), так и отдельные их элементы (например, %o1[1][1]).

В случае, когда уравнений меньше, чем неизвестных, solve поступит точно так же, как и в случае одного уравнения с несколькими символами: все неуказанные будет воспринимать как параметры:

(%i1) $\mathrm{solve}([x^2+y^2=a^2, x+y=2a+1], [x, y])$

(%o1) $$\left[\left[x=-\frac{\sqrt{-2a^2-4a-1}-2a-1}{2}, y=\frac{\sqrt{-2a^2-4a-1}+2a+1}{2}\right], \left[x=\frac{\sqrt{-2a^2-4a-1}+2a+1}{2}, y=-\frac{\sqrt{-2a^2-4a-1}-2a-1}{2}\right]\right]$$

Если solve не находит точных решений, она может, как и integrate, вернуть уравнение или систему уравнений в некотором упрощенном виде, а может и самостоятельно попытаться решить систему численно:

(%i1) $\mathrm{eqs}: [4x^2-y^2=12, x\,y-x=2]\$$

(%i2) $\mathrm{solve}(\mathrm{eqs})$

(%o2) $[[y=2, x=2], [y=-0.15356757100197, x=-1.733751846381093], [y=3.60800322187\backslash 0287\,i+0.076783785237878, x=-0.5202594388652\,i-0.13312403573587], [y=0.07678378523\backslash 7878-3.608003221870287\,i, x=0.5202594388652\,i-0.13312403573587]]$

В таком случае, если вам все же нужны точные значения корней (в аналитической записи), либо если они не найдены даже в числах, можно попробовать решить уравнения по очереди, выражая одно неизвестное через другое:

(%i3) $\mathrm{solve}(\mathrm{eqs}[2], y)$

(%o3) $$\left[y=\frac{x+2}{x}\right]$$

(%i4) $\mathrm{eqs}[1], \%$

(%o4) $$4x^2-\frac{(x+2)^2}{x^2}=12$$

(%i5) $\mathrm{solve}(\%)$

И подставляя в оставшиеся уравнения:

(%i5) $\mathrm{solve}(\%)$

(%o5) $$\left[x=\frac{7\left(\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)}{36\left(\frac{\sqrt{139}}{24\sqrt{3}}-\frac{8}{27}\right)^{\frac{1}{3}}}+\left(\frac{\sqrt{139}}{24\sqrt{3}}-\frac{8}{27}\right)^{\frac{1}{3}}\left(-\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)-\frac{2}{3}, x=\left(\frac{\sqrt{139}}{24\sqrt{3}}-\frac{8}{27}\right)^{\frac{1}{3}}\left(\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)+\frac{7\left(-\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)}{36\left(\frac{\sqrt{139}}{24\sqrt{3}}-\frac{8}{27}\right)^{\frac{1}{3}}}-\frac{2}{3}, x=\left(\frac{\sqrt{139}}{24\sqrt{3}}-\frac{8}{27}\right)^{\frac{1}{3}}+\frac{7}{36\left(\frac{\sqrt{139}}{24\sqrt{3}}-\frac{8}{27}\right)^{\frac{1}{3}}}-\frac{2}{3}, x=2\right]$$

Теперь можем подставить обратно – и найти значения второй неизвестной, например, для первого и последнего корней из последнего списка:

(%i6) $\%o3[1], \%[1]; \%o3[1], \%th(2)[4]$

(%o6) $$y=\frac{\frac{7\left(\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)}{36\left(\frac{\sqrt{139}}{24\sqrt{3}}-\frac{8}{27}\right)^{\frac{1}{3}}}+\left(\frac{\sqrt{139}}{24\sqrt{3}}-\frac{8}{27}\right)^{\frac{1}{3}}\left(-\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)+\frac{4}{3}}{\frac{7\left(\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)}{36\left(\frac{\sqrt{139}}{24\sqrt{3}}-\frac{8}{27}\right)^{\frac{1}{3}}}+\left(\frac{\sqrt{139}}{24\sqrt{3}}-\frac{8}{27}\right)^{\frac{1}{3}}\left(-\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right)-\frac{2}{3}}$$

(%o7) $y=2$

Функция solve имеет довольно большое количество различных переключателей, из которых может пригодиться в своем не-умолчательном значении в первую очередь один: solveradcan. Умолчание здесь равно ***false***, а выставив этот флаг в ***true***, мы заставим solve, помимо его умолчательного поведения, применять radcan – функцию по упрощению показательных, логарифмических и степенных (с рациональными степенями) функций. Это делает работу функции solve более медленной (потому по умолчанию этот режим и выключен), но в некоторых случаях может помочь разрешить проблемы, которые без этого ключа приведут к невозможности найти точное решение.

И снова дифференцируем и интегрируем.

Помимо «просто» уравнений, *Maxima* позволяет также решать и обыкновенные дифференциальные уравнения первого и второго порядка. Функций, непосредственно занимающихся решением таких уравнений, существует две. Первая из них занимается поиском частных решений линейных дифференциальных уравнений и систем таких уравнений; зовут ее desolve, от слов «differential equation solve». Эта функция принимает два аргумента, первый из которых – уравнение либо список уравнений, а второй – соответственно одна переменная или список переменных. Если не заданы значения функций и/или их производных в нуле, то в найденном решении они просто отображаются в виде f(0) или

$$\left.\frac{d}{dx}f(x)\right|_{x=0}$$

Задать эти значения позволяет функция atvalue(выражение, переменная = точка, значение); то есть, в данном случае atvalue(f(x), x=0, значение) или atvalue('diff(f(x)), x=0, значение). Производные в уравнениях и системах, решаемых с помощью этой функции, должны быть записаны непременно в виде 'diff(f(x), x), а не просто 'diff(f, x), а сами функции, соответственно, тоже в виде f(x), а не f – нужно продемонстрировать зависимость функции от ее аргумента.

(%i1) $['\mathrm{diff}(f(x),x)='\mathrm{diff}(g(x),x)+\sin(x), '\mathrm{diff}(g(x),x,2)='\mathrm{diff}(f(x),x)-\cos(x)]$

(%o1) $$\left[\frac{d}{dx}f(x)=\frac{d}{dx}g(x)+\sin x, \frac{d^2}{dx^2}g(x)=\frac{d}{dx}f(x)-\cos x\right]$$

(%i2) $\mathrm{atvalue}('\mathrm{diff}(g(x),x), x=0, a)\$ \; \mathrm{atvalue}(f(x), x=0, 1)\$$

(%i4) $\mathrm{desolve}(\%o1, [f(x), g(x)])$

(%o4) $[f(x)=a\,e^x-a+1, g(x)=\cos x+a\,e^x-a+g(0)-1]$

И конечно же, точно так же как для обычных уравнений и систем, здесь мы тоже можем проверить решение с помощью подстановки, но

только надо еще дополнительно задать принудительное вычисление производных, так как в уравнениях они фигурируют в несовершенной форме:

(%i5) $\%o1, \%, \mathrm{diff}$

(%o5) $[a\,e^{x}=a\,e^{x}, a\,e^{x}-\cos x=a\,e^{x}-\cos x]$

Вторая функция из этой группы называется ode2 и предназначена она для решения обыкновенных дифференциальных уравнений первого и второго порядка; ее название происходит от фразы «ordinary differential equations of 1st or 2nd order». Пишется она так: ode2(уравнение, зависимая-переменная, независимая-переменная). Здесь уже независимая переменная указывается в списке параметров функции явно, и потому обозначения вида y(x) не нужны: и функция, и переменная обозначаются просто одиночными буквами. Также в отличие от предыдущей функции, ode2 ищет не частное, а общее решение. Произвольная константа в решении уравнения первого порядка обозначена через %c; в решении уравнения второго порядка таких констант, естественно, две, и обозначаются они как %k1 и %k2.

(%i1) $x^2\,'\mathrm{diff}(y,x)+3\,y\,x=\dfrac{\sin(x)}{x}$

(%o1) $x^2\left(\dfrac{d}{d\,x}\,y\right)+3\,x\,y=\dfrac{\sin x}{x}$

(%i2) $\mathrm{ode2}(\%, y, x)$

(%o2) $y=\dfrac{\%c-\cos x}{x^3}$

(%i3) $'\mathrm{diff}(y,x,2)+y\,'\mathrm{diff}(y,x)^3=0$

(%o3) $\dfrac{d^2}{d\,x^2}\,y+y\left(\dfrac{d}{d\,x}\,y\right)^3=0$

(%i4) $\mathrm{ode2}(\%, y, x)$

(%o4) $\dfrac{y^3+6\,\%k1\,y}{6}=x+\%k2$

В дополнение к функции ode2 существуют три функции для поиска частных решений на основе полученных общих. Иначе говоря, эти функции, получая конкретные условия относительно значения функции-решения в заданной точке, находят исходя из этих значений соответствующие им величины интегральных констант. Одна из этих функций предназначена для обработки решения дифференциального уравнения первого порядка. Она называется ic1 (i от «initial value» – начальное значение; c от «constant» – константа; 1 от «1st order» – первого порядка) и принимает три аргумента: первый – само решение, в том виде, в котором его находит функция ode2; второй – значение независимой переменной (x-координаты), третий – значение функции (зависимой переменной, y) при этом значении x и возвращает частное решение, проходящее через точку с заданными координатами (x, y):

(%i5) $\mathrm{ic1}(\%o2, x=\pi, y=0)$

(%o5) $y=-\dfrac{\cos x+1}{x^3}$

И две функции работают с решениями уравнений второго порядка. Так как в общем решении уравнения второго порядка фигурируют две независимые константы, то эти функции задают уже по два условия для поиска частного решения. Первая функция выглядит как ic2(общее решение, x, функция-в-точке-x, производная-в-точке-x). Расшифровка названия аналогична предыдущей функции. Действует тоже аналогично ей, а в качестве второго условия задает значение производной в той же заданной точке:

(%i6) $\mathrm{ic2}(\%o4, x=0, y=0, '\mathrm{diff}(y,x)=2)$

(%o6) $\dfrac{y^3-3\,y\,(y^2-1)}{6}=x$

(%i7) $\mathrm{ratsimp}(\%)$

(%o7) $-\dfrac{2\,y^3-3\,y}{6}=x$

И последняя функция называется bc2 (b от «boundary value» – граничное значение; а дальше все как в предыдущей). Ее аргументы: первым, как и в двух остальных вариантах, идет само общее решение, возвращенное функцией ode2; после него идут две пары значений: x0, y0, x1, y1, задающие две точки, через которые должен проходить график функции-решения:

(%i8) $\mathrm{bc2}(\%o4, x=0, y=1, x=1, y=3)$

(%o8) $\dfrac{y^3-10\,y}{6}=x-\dfrac{3}{2}$

Касательно дифференциальных уравнений все этими функциями и заканчивается, никаких дополнительных ключей к ним не предназначено.

В следующий раз мы поговорим о средствах построения графиков функций и о средствах автоматизации: условных операторах, циклах и управляющих условиях, налагаемых на различные символы и выражения. LXF



Через месяц Графики, управляющие структуры и циклы.



Ответы

Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru

1 Без паники...

В Я установил игру *Pipepanic* с DVD к *LinuxFormat* [HotPicks, LXF82] в свою домашнюю директорию и запускаю ее, набрав **./pipepanic** в консоли:

```
cd Pipepanic-0.1.3-source
./Pipepanic
```

Но не могу добавить соответствующий пункт в меню KDE, поправив меню в Центре Управления KDE: не знаю, что набирать в поле команды. Если я помещаю туда **/home/marrea/pipepanic-0.1.3-source/pipepanic**, а затем выхожу и щелкаю на добавившемся пункте меню *Pipepanic*, то вижу лишь кувыркание песочных часов и кнопку на панели задач. Секунд через 30 все исчезает. Это потому, что я установил *Pipepanic* в свою домашнюю директорию?

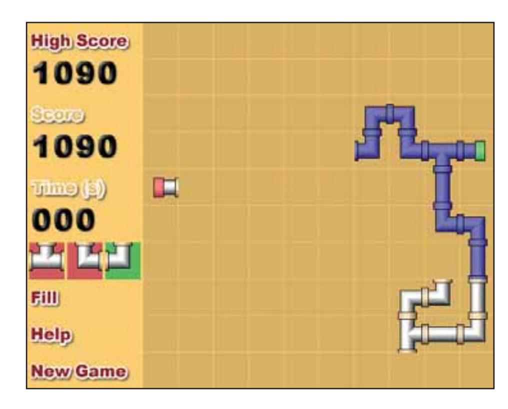
Marrea, с форума LXF

О Дело вот в чем: запуская игру из меню KDE, Вы находитесь не в директории **Pipepanic**. Программа должна запускаться из своей директории, чтобы найти нужные ей файлы. Это можно исправить, добавив **/home/marrea/Pipepanic-0.1.3-source/** в поле рабочего каталога в редакторе меню KDE – по сути, добавив команду смены каталога перед запуском, которую вы выполняете в консоли. Возможно, также понадобится указать полный путь к файлу. Самый безопасный способ убедиться, что все правильно – воспользоваться диалогами выбора файлов, вызываемых по щелчку на пиктограммах справа от соответствующих полей.

Если вы установите опцию запуска в терминале, вы увидите весь вывод программы и, возможно, поймете, в чем проблема. Именно так я узнал, что игре не удалось найти файл. В конец запускаемой команды добавьте что-нибудь вроде ; sleep 5, чтобы окно терминала оставалось открытым еще пять секунд после завершения программы:

```
/home/marrea/Pipepanic-0.1.3-source/Pipepanic; sleep 5
```

НБ



› **Запускайте *Pipepanic* из меню KDE, предварительно указав его директорию.**

2 Скрупулезный скрипт

В Попытался написать скрипт для резервирования важных данных из домашней директории на USB-диск с помощью rsync, и теперь нахожусь в затруднении: озадачили опции --exclude-from и особенно --include-from. У меня Ubuntu 6.06 и rsync 2.6.6.

Вот приблизительно то, что мне нужно. Во-первых, скопировать нескрытые файлы и директории (включая вложенные), находящиеся в домашнем каталоге **/home/dcoldric**; исключение – **/home/dcoldric/MyDownloads**, где поддиректории копировать не надо, только обычные файлы. Еще одно исключение – не хочу резервировать кое-какие нескрытые каталоги, вроде **/home/dcoldric/cxoffice**.

Все скрытые файлы и директории должны быть проигнорированы, кроме нескольких. Например, я хочу сохранить **/home/dcoldric/.netbeans** со всеми поддиректориями, а также **.bashrc** и **.bash_aliases**. И, наконец, я бы хотел, чтобы структура директорий каталога резервных копий была такой же, как и в моем домашнем каталоге (не считая, конечно, игнорируемых директорий).

Я перепробовал все, что пришло в голову, но безрезультатно. Мой последний вариант выглядел так:

```
rsync -a --delete --safe-links --exclude-from=/
  home/dcoldric/bin/backupExcludes /home/
  dcoldric/ /media/USB/backup/dcoldric
```

– при следующем содержимом файла **backupExcludes**:

```
- /*
+ /dcoldric/
+ /dcoldric/.Creator/
+ /dcoldric/.java/
+ /dcoldric/.mozilla/
+ /dcoldric/.mozilla-thunderbird/
+ /dcoldric/.netbeans/
+ /dcoldric/.bashrc
+ /dcoldric/.bash_aliases
+ /dcoldric/MyDownloads/
- /dcoldric/MyDownloads/*/
- /dcoldric/.*
- /dcoldric/cxoffice
- /dcoldric/jdk*
- /dcoldric/sun
-/dcoldric/SUNW*
```

Однако ничего не работает.

Дэвид Колдрик [David Coldrick]

Наши эксперты

» **Мы найдем эксперта на любой вопрос! Вы получите ответ на все: от проблем с установкой или модемом до сетевого администрирования; главное – спросить!**



Нейл Ботвик
Попробовав себя в роли провайдера, а также редактора дисков *Linux Format*, Нэйл по праву именует себя мастером на все руки.



Валентин Синицын
В свободное от работы время редактор нашего журнала разрабатывает KNetworkManager и другие открытые KDE-приложения. Он с радостью поможет вам в вопросах использования Linux на рабочем столе.



Кингс Кобблер
Кингс – системный инженер Linux и администратор Rackspace, использует Linux десять лет, всегда готов отвечать на письма сисадминов.

КУДА ПОСЫЛАТЬ ВОПРОСЫ:
Пишите на м по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru

»

» **О** Команда rsync по умолчанию копирует все, а опция --exclude задает то, что нужно пропустить. Проще думать, что --include – это --do-not-exclude. Указываемый параметр-файл --exclude-from – фактически файл фильтра. Фильтрация предоставляет контроль, но не имеет варианта --filter-from. Вот более корректный способ использовать фильтр:

```
--filter="merge myfilterfile"
```

Ваш текущий фильтр не работает, поскольку начинается с -/*, то есть с указания игнорировать все. Вы говорите, что ничего не происходит – и Вы правы, но программа тоже права, поскольку делает то, что ей велено. Так как начинает она с первого вхождения, то перенесите -/* в конец.

Когда фильтр начинается с /, он считает от указанного исходного пути (в данном случае получается **~/dcoldric**), так что удалите /dcoldric из начала каждой строки, иначе rsync припишет это лишний раз и будет искать **/home/dcoldric/dcoldric/.mozilla** и так далее.

Хотя это не относится к Вашему текущему фильтру, следует знать, что

```
+ /foo/bar/
- /*
```

ничего не найдет, поскольку /* исключит все каталоги из исходной директории и foo не будет проверяться никогда, значит, **/foo/bar** не будет найден. Вы должны явно указать сканирование **foo**:

```
+ /foo/
+ /foo/bar/
- /foo/*
- /*
```

Работающий файл фильтра будет выглядеть так:

```
+ /.netbeans/
+ /.bashrc
+ /.bash_aliases
- /MyDownloads/*/
- /.*
- /cxoffice
- /jdk*
```

Вызывается он такой командой:

```
rsync -a --delete --safe-links --filter="merge ~dcoldric/bin/backupFilter" ~dcoldric/ /media/USB/backup/dcoldric/
```

Обратите внимание на завершающий каталог назначения /, он может повлиять на результат. **НБ**

3 Которая Fedora?

В Я купил Fedora Core 4 и специальный выпуск *LinuxFormat* в 2005 году, и теперь хочу установить Fedora на *VMWare* (под Windows XP). В *VMWare* есть несколько альтернатив для Red Hat, например, Red Hat Linux и Red Hat Enterprise Linux 2,3 и 4.

Я подозреваю, что могу отбросить чистый Red Hat Linux, но какой из других вариантов подойдет для моей версии Fedora? По идее, это важно, поскольку *VMWare Tools* очень улучшают работу видеокарты и мыши в гостевой ОС. Но во всех дистрибутивах, которые я пробовал, этого не произошло, и я не могу найти нужной информации в журнале.

Изн Стенд [Ian Stand]

О Почти любой дистрибутив Linux, который я пробовал устанавливать в *VMWare* (а я пробовал много), инсталлировался нормально, даже если не был указан в списке. В большинстве случаев я выбирал Other Linux 2.6.x kernel, но для Fedora Core я выбрал «чистый» Red Hat Linux. Установка *VMWare Tools* описана на стр. 142 руководства по *VMWare Workstation* (его можно скачать с www.VMWare.com/support/pubs/ws_pubs.html) и пройдет без проблем, если следовать инструкциям:

1. **Удалите все смонтированные CD/DVD-диски.**
2. **Выберите VM > Install VMWare Tools из меню VMWare.**
3. **Откройте CD-ROM в гостевой ОС.**
4. **Дважды щелкните по RPM-файлу VMWare-Tools.**
5. **Введите пароль root.**
6. **Запустите VMWare-config-Tools.pl в консоли от имени root.**



› **Даже Windows-машина может приютить Linux, благодаря *VMWare* и Fedora Core 4.**

Для финального шага Вам может понадобиться установленный *GCC*, если потребуется сборка модуля для Вашего ядра. Это произойдет, если установщик не найдет уже готового модуля, как, например, для Fedora Core 5. **НБ**

4 Apache защищается

В Я недавно завел web-сервер на *Apache* для обслуживания нескольких сайтов. Можете ли вы мне дать несколько советов по укреплению его безопасности?

С.Монро [S. Monroe]

О Кроме защиты самих страниц с помощью HTTP-аутентификации или SSL, где только можно, кое-что надо сделать в файле **http.conf**, так как настройки по умолчанию могут помочь атакующему, предоставив ему некоторую специфическую информацию. Во-первых, убедитесь, что установили директорию ServerTokens в Prod. По умолчанию, этот параметр будет отображать номер Вашей версии *Apache*, состав используемых модулей и даже ОС. Хотя в пределах своей вотчины тут скрывать нечего, Вы вряд ли захотите сообщать такие данные посторонним лицам [эта рекомендация затруднит получение информации об ОС, но не сделает ее полностью невозможной, – прим. ред.].

Чтобы посмотреть, что именно показывает Ваш сервер, выполните

```
curl -I http://yourwebserver
```

Также убедитесь, что ServerSignature установлена в email – это предотвратит «засветку» номера версии на страницах ошибок *Apache*. Хотите ли Вы предоставить Вашим пользователям собственные web-папки? Нет? Тогда отключите модуль userdir. Если Вы не »



Краткая справка по...

Top

Удостоверьтесь, что ресурсы системы распределены справедливо.

Казалось ли вам когда-нибудь, что ваш компьютер еле ползает, не реагирует на ваши действия или вгоняет в клинч жесткий диск? Если такое случится опять, наберите в терминале **top**. Эта программа вмещает много информации в малую площадь. По умолчанию в верхней строке отображается: время работы системы (uptime), число работающих пользователей, число выполняющихся задач и так далее. Здесь также отображается использование процессорного времени для пользовательских задач (us), системных операций (sy), неприоритетных задач (ni), бездействующих (id) и ожидающих (wa) процессов. Бездействие показывает, какой запас процессорных ресурсов у вас еще есть. Ожидающие процессы – это такие программы, которые ждут каких-то событий (обычно ввода/вывода).

На рисунке показан список задач, отсортированный по использованию процессорного времени – тут легко определить, кто сожрал львиную долю циклов вашего CPU. Для смены критериев сортировки, например, по использованию памяти, нажимайте F.

Top не просто отображает ваши процессы: вы можете влиять на их работу. Нажмите k, чтобы убить процесс, r – чтобы понизить его приоритет, если он отбирает слишком много ресурсов. Понизить приоритет – все равно, что сказать процессу «будь повежливей, пропускай других вперед». Убийство и понижение приоритета требуют указания идентификатора процесса (PID), помещенного в левой колонке. Чтобы по ошибке не угробить невиновного, после набора **top** в X-терминале просто щелкните дважды на нужном ID и вставьте его средней кнопкой мыши.

› ***Top* показывает, сколько используется памяти и циклов процессора.**

У *top* есть множество опций, большинство из которых имеют горячую клавишу. Самая нужная клавиша – h, она выводит справочную информацию.

[Вы также можете попробовать htop – улучшенную версию стандартного top, доступную по адресу htop.sourceforge.net, – прим. ред]

» используете CGI, удалите псевдоним **cgi-bin** из файла конфигурации.

Еще одна опасная штука – руководство по *Apache*, которое в некоторых сборках установлено как страница по умолчанию. Убедитесь, что у вас запрещены листинги каталогов, установив **–Indexes** в секции Options директив <Directory>. Если у Вас работает PHP, убедитесь, что директива *expose_php* в **php.ini** отключена (Off).

Если другие люди публикуют информацию на Вашем web-сервере, лучше позаботиться, чтобы они не могли менять настройки сервера через .htaccess. Для этого в пределах корневой директивы <Directory> установите параметр AllowOverride в None, AuthConfig или другое ограниченное значение, но ни в коем случае не All. **КК**

5 Терминал последовательно

В Я поддерживаю некоторое древнее производственное оборудование, и у меня есть несколько тестовых программ, я написал их много лет назад на QuickBasic. Мониторинг результатов теста производит *HyperTerminal*, настроенный на эмуляцию DEC VT100 на COM1 (9600 бод). Еще я использую анализатор Thurlby LA160 и осциллограф Velleman. Все это работает под Windows.

Не расскажете ли вы, как получить такой же VT100 на Linux? Нужно ли мне изучить *Wine* для запуска Thurbly и Velleman под Linux? И что будет с моими QuickBasic-программами (откомпилированными)? У меня сейчас установлены Windows ME и SUSE 10.0.

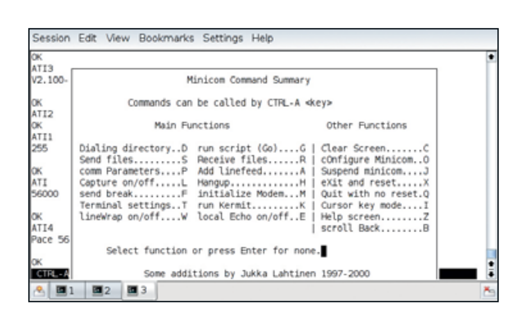
Алан Киббл [Alan Kibble]

О Последовательные порты в Linux обозначаются, начиная с **/dev/ttyS0**, который соответствует COM1. Возможно, у Вас даже настроена ссылка с **/dev/modem** на **/dev/ttyS0**. Вместо *HyperTerminal* обычно используют *Minicom*, доступный для множества дистрибутивов, включая SUSE 10.0. *Minicom* имеет режим эмуляции VT100, как раз он Вам и нужен. Пакет SUSE не имеет глобальных настроек, так что после установки надо сперва запустить

```
minicom -s
```

от имени суперпользователя. Чтобы иметь право записи в последовательные устройства, Вы также должны входить в группу UUCP; можно это настроить в Yast > Security And Users > User Management, но изменения войдут в силу только после перезапуска KDE.

Для запуска любого проприетарного ПО Вам, скорее всего, потребуется *Wine*, и вместо COM1 для доступа к оборудованию будет использоваться **/dev/ttyS0**. Ваше ПО на QuickBasic тоже заработает под *Wine*, но в долгосрочной перспективе будет лучше портировать его на что-нибудь вроде Gambas (Linux-эквивалент Visual Basic) или какой-либо кросс-платформенный язык типа Python. **НБ**



› ***Minicom*** **– стандартный последовательный терминал с эмуляцией VT100.**

6 Захват пространства

В Мне нужно добавить пространства подкачки в мою Linux-систему, но неразмеченного места на диске больше нет. Что тут можно сделать?

И. Уоллис [I Wallis]

О GNU/Linux значительно превосходит другие ОС гибкостью, в том числе и по части управления памятью. Во-первых, определите, какого размера файл подкачки Вам нужен – допустим, надо добавить гигабайт. Далее выберите раздел, где достаточно свободного места. На моей машине, например, в **/opt**-разделе размером 4 ГБ занято только 1,5 ГБ.

Теперь настало время создать файл, который будет использоваться для подкачки. Для этого вам нужна команда dd, среди ее аргументов – размер блока и их количество. Для создания гигабайтного файла используйте такую команду:

```
dd if=/dev/zero of=/opt/swapfile bs=1G count=1
```

Эта команда создаст файл размером в 1 ГБ и запишет его в **/opt/swapspace**. Параметр if указывает источник записи, **of** – файл назначения. Отформатируйте этот файл:

```
mkswap /opt/swapfile
```

После этого можете активировать его следующей командой:

```
swapon /opt/swapfile
```

Активные источники подкачки вы можете увидеть, »

Часто задаваемые вопросы...

Шифрование файлов

Способы сделать ваши файлы недоступными.

» Опасаюсь, что правительство влезет в мой компьютер и изымет у меня свидетельства высадки инопланетян. Как мне себя защитить?

Можете зашифровать файлы с помощью, например, *GnuPG*. Тогда содержимое файлов будет недоступным, пока вы его не дешифруете с помощью пароля. *GnuPG*, или *GPG* – *GPL*-заменитель *PGP*, стандарта де-факто для шифрования файлов.

» Нет ли способа попроще, чтобы обезопасить файлы?

Есть такой: с помощью зашифрованной файловой системы. Она автоматически шифрует файлы при записи и дешифрует при чтении. Вы набираете пароль только один раз, когда монтируется сама файловая система. Если ваш ноутбук будет потерян или украден, никто не сможет прочесть информацию с зашифрованных файловых систем (обычно это **/home**).

» Почему только /home? А все зашифровать нельзя?

Нет, потому что будет невозможно получить доступ к ПО, необходимому для монтирования разделов. Да и какой смысл зашифровывать системные директории, если там только пакеты с бесплатно загружаемых дисков?

» И что мне надо сделать?

Тут есть альтернативы. Ядро Linux содержит **dm-crypt**, который можно использовать для создания полностью зашифрованных файловых систем. Имеется несколько способов управления созданием и монтированием таких файловых систем. Я предпочитаю LUKS (*Linux Unified Key Setup*); другой вариант – EncFS. Это файловая система FUSE (см. **LXF76**). Вместо шифрования всей файловой системы, она шифрует индивидуальные файлы: их имена и содержимое. Это не самый безопасный вариант, поскольку размеры файлов и их права доступа все-таки видимы, но он более удобен.



› **EncFS дает пользователю контроль над данными на жестком диске или USB-брелке.**

» А как насчет резервных копий?

Если вы шифруете всю файловую систему, нужно смонтировать ее перед снятием копий; для защиты резервных копий зашифруйте их с помощью GPG. Когда вы используете EncFS, сама файловая система стандартна, зато содержимое файлов выглядит как мусор. Можете использовать любую программу для сохранения их резервных копий, файлы останутся зашифрованными. Это удобно, если резервные копии хранятся на удаленном сервере.

» Это спасет файлы, если я потеряю свой ноутбук?

Увы, нет, если его похитят спящим (suspend) при неразмонтированной зашифрованной файловой системе. EncFS автоматически выполняет размонтирование файловой системы, если она какое-то время не используется – но только при отсутствии открытых файлов. Если для шифрования раздела, например, **/usr**, вы используете **dm-crypt**, нужно выйти из системы и размонтировать **/usr** перед переводом в режим спячки.

» выполнив команду cat /proc/swaps или free. Чтобы подкачка подключалась при загрузке системы автоматически, добавьте в **/etc/fstab** соответствующую строку:

```
/opt/swap swap defaults 0 0
```

КК

7 ADSL

В Я купил *LinuxFormat* с диском Mepis [LXF79], потому что он превосходно определяет оборудование, а у меня с этим проблемы. К сожалению, соединение с Интернет не работает и здесь. Можете ли вы подсказать, как заставить работать ADSL в Linux? У меня Xircom Creditcard Ethernet 10/100 + Modem 56. В настройках соединения я нашел, что адрес дожен присваиваться по DHCP. Попытался скопировать другие настройки, включая IP-адреса, маска подсети, шлюз и DNS – ничего не помогает.

Джимми Янссен ван Раай [Jimmy Janssen van Raay]

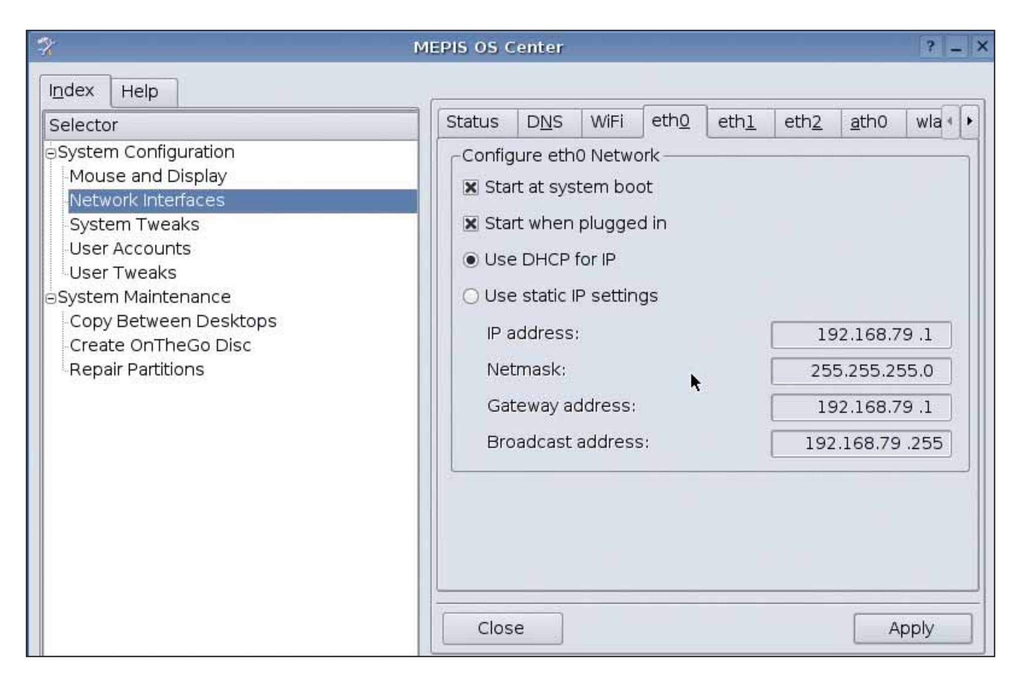
О Сетевая сторона этой карты обрабатывается модулем **xirc2ps-cs**, в Mepis он включен. Во-первых, проверьте, была ли обнаружена карта. Откройте терминал и наберите

```
su -
# введите пароль root
lsmod | grep xirc
```

Если вывод пуст, значит, модуль не загружен, и надо его загрузить:

```
modprobe xirc2ps-cs
```

На сей раз пустой вывод будет значить, что все в порядке. Теперь запустите



› **OS Centre в Mepis – это такое место, где можно настроить все, причем не обращаясь к терминалу.**

```
ifconfig
```

для просмотра ваших сетевых интерфейсов. Их должно быть два: lo и ваш сетевой интерфейс, скорее всего eth0.

Запустите Mepis OS Centre из меню KDE, перейдите в секцию Network и выберите ваш сетевой интерфейс. Во вкладке DNS укажите Use DHCP For IP, а также выберите DHCP на вкладке DNS. Теперь Ваша сеть должна настроиться автоматически. Если карта при загрузке системы не запускается сама, выполните из-под root такую команду:

```
echo "xirc2ps-cs" >>/etc/modules
```

Она добавит имя модуля в список автозагружаемых. НБ

Вопрос-победитель

Дмитрий Худорожков получает подарочный сертификат на 1000 рублей от интернет-магазина LinuxCenter.Ru! Мы поздравляем победителя и просим его связаться с редакцией: info@linuxformat.ru

★ Украшаем Gnome

В Подскажите, пожалуйста, можно ли в GNOME сделать так, чтобы фон рабочего стола регулярно изменялся (в KDE такая функция называется "Слайд-шоу")? И нельзя ли извлечь иконки из EXE-файлов Windows, чтобы затем использовать их как значки приложений в GNOME? Спасибо

Дмитрий Худорожков

О Насколько нам известно, встроенной функции слайд-шоу в Gnome нет, но это не значит, что ее нельзя смоделировать – например, простым shell-скриптом. Ключевой момент – вызов **gconftool** для обновления фона рабочего стола, все остальное вы легко можете подстроить под свои нужды. Самый простой сценарий может выглядеть следующим образом:

```
#!/usr/bin/bash
file_list=$1
delay=$2
while read picture; do
gconftool-2 --direct --config-source xml:readwrite:/etc/gconf/gconf.xml.mandatory --type string --set /desktop/gnome/background/picture_filename $picture
sleep $delay
done < $file_list
```

Сохраните его куда-нибудь, например, в **/usr/local/bin/gslideshow** (потребуются права root) и сделайте исполняемым:

```
chmod +x /usr/local/bin/gslideshow
```

Скрипт принимает два параметра - имя файла со списком картинок и задержку в секундах. Файл-список в простейшем случае можно сгенерировать командой **ls**:

```
ls <каталог с картинками> | sort >~/backgrounds.lst
```

Тогда команда вызова скрипта может иметь следующий вид

```
/usr/local/bin/gslideshow /home/user/backgrounds.lst 60
```

Более продвинутый вариант сценария можно найти на форуме Ubuntu: http://ubuntuforums.org/archive/index.php/t-2801.html

Что же касается пиктограмм, сохраненных в ресурсах EXE-файлов, рекомендуем Вам извлечь их при помощи любого редактора ресурсов Windows (подойдет даже тот, что входит в состав MinGW) и затем сохранить в формате ICO – с ним работать гораздо проще. ВС

8 Какойтотам.rpm

В У меня два вопроса, и оба касаются забытых имен программ.

Первый касается удаления файлов по собственной глупости (компьютеры делают не то, что вы хотите, а то, что вы велите им сделать, и мне надо защититься от собственных ошибок). Я когда-то читал (может, даже и в вашем журнале) про демон *undelete*. Все команды удаления файлов перехватываются – наверное, переопределен системный вызов unlink или что-то вроде – и файлы вместо удаления перемещаются в корзину. А потом оказывается, что немедленно очистить корзину, как большинство и делает, вы не можете. Файлы не удаляются, пока размер корзины не превысит определенного порога или пока объем свободной памяти не упадет ниже другого порога.

Второй – я web-разработчик, и мне нужно тестировать страницы на как можно большем количестве браузеров. Я слышал о проекте браузера *KHTML* на *GTK*, он бы мне очень пригодился, да не могу вспомнить имя проекта. С *Konqueror* связываться не хочется: придется целиком загружать KDE, а полнофункциональный браузер мне ни к чему. Хватило бы и легкой альтернативы.

Дэйв Спейси [Dave Spacey]

О Что касается первой программы – это, наверное, *Delsafe* (http://delsafe.cjb.net). Она работает в точности как Вы описали, подменяя удаление файлов на перемещение их в корзину. Многократные удаления файла с одним и тем же именем ассоциируются с отметками времени, а программа *undel* восстанавливает файлы. Другой вариант – *libtrash* (http://pages.stern.nyu.edu/~marriaga/software/libtrash), имеющая похожие функции.

Я подозреваю, что, говоря про проект KHTML, Вы имели в виду *GTK+ Webcore* (http://GTK-webcore.sourceforge.net). Этот проект пока сыроват, и вряд ли он Вам подойдет. Я бы все-таки посоветовал взять *Konqueror*, особенно если на Ваших страницах используется JavaScript. Не так все плохо: устанавливать весь KDE не потребуется, нужны лишь пакет *kdelibs* и сам *Konqueror*. Во многих дистрибутивах пакет для KDE теперь расщеплен, и достаточно установить один *Konqueror* вместо всего *kdebase*, как было раньше. **НБ**

9 О чернухе

В Мой вопрос к вам по поводу черных списков *DansGuardian*. Что ни день появляется несметное количество сайтов, и системный администратор должен периодически обновлять эти списки. Не у всех хватает средств на оплату такой работы. Можно ли обновлять списки автоматически с помощью web-пауков? А если да, то как это сделать, что моя система может из-за этого выиграть или проиграть? Не могли вы привести необходимый минимум требований, чтоб запускать *DansGuardian* без проблем?

Гбенга Дада [Gbenga Dada]

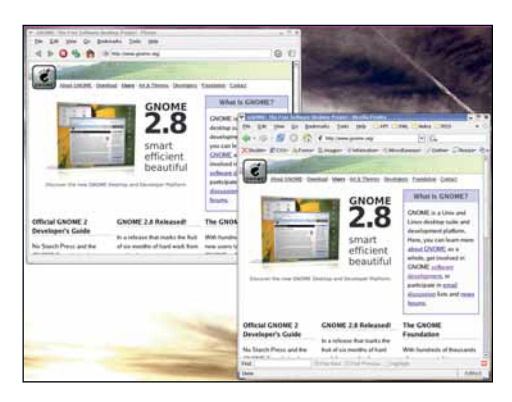
PS: И что нужно делать, чтобы основать группу пользователей Linux (LUG) в Нигерии?

О Прежде всего учтите: *DansGuardian* занимается не только черными списками. Это фильтр содержимого, в основном работающий с контентом страниц. И он время от времени обновляет свой список «нехороших» фраз, так как создатели сайтов постоянно норовят обойти фильтрацию, а заодно волей-неволей и Ваш список обновит. Свежий вариант списка находится на http://contentfilter.futuragts.com/phraselists.

Использование пауков для обновления черных списков URL было бы очень затратно по трафику, поскольку они будут проверять и те сайты, которые Вы никогда не посетили бы, причем используя один и тот же список фраз. Можно скачивать обновляемые списки URL – некоторые из них распространяются на коммерческой основе, некоторые бесплатны. Коммерческие списки иногда просто склеивают несколько бесплатных; Вы платите за то, чтоб самим этого не делать. На сайте *DansGuardian* (секции xtras и Add-ons) есть скрипты, умеющие скачивать и устанавливать последние версии списков. Также Вы можете получить их с сайта *Squidguard*: www.squidguard.org/blacklist.

Рекомендации по запуску программы зависят от ее использования. Для домашней сети требования минимальны. Основная нагрузка на систему происходит при начальной загрузке правил, так что большой объем памяти важнее быстрого процессора. Влияет также состав задач, работающих на Вашей машине.

Все, что нужно для создания группы пользователей Linux – это люди и место встречи или сайт и список рассылки, если Ваша группа существует только в киберпространстве. Никаких формальных требований нет, нужны лишь люди, разделяющие Ваш интерес к Linux. Одни группы устраивают официальные встречи, с докладами участников, другие просто собираются поболтать про Linux, да и не только про Linux. Вы найдете их уставы на сайтах http://en.tldp.org/HOWTO/User-Group-HOWTO.html и http://linuxmafia.com/faq/Linux_PR/newlug.html. **НБ**



› **Проект *GTK+Webcore* разрабатывает версию KHTML, использующую *GTK*.**

10 Плохая опция

В Я как-то попытался смонтировать один из моих внешних дисков, и получил следующее сообщение об ошибке:

```
mount: wrong fs type, bad option, bad superblock on /
dev/hda1, or too many mounted file systems
```

Проверив диск с помощью fsck, я получил вот что:

```
fsck.ext3: No such file or directory while trying to open
/dev/hda1
```

Суперблок не читается и не описывает правильную файловую систему ext2.

Диск я, допустим, заменю, но данные с него хотелось бы восстановить. Возможно ли это?

Д.Клингшотт [D Klingshott]

О К счастью, да! Файловые системы ext2 и ext3 создают резервные суперблоки, записывая их через определенные интервалы на диск, Вам просто нужно их найти и указать на них fsck после починки файловой системы. Местоположение суперблоков зависит от размера раздела. Простейший способ их обнаружить – перезапустить *mke2fs* с ключом **-n**. Делать *mke2fs* ничего не будет, но сообщит, что бы он предпринял:

```
mke2fs /dev/hda
```

Данная команда среди прочего выведет список суперблоков в файловой системе. Используя эту информацию, вы можете при помощи *fsck* восстановить суперблок с одного из этих мест.

```
fsck -b 8193 /dev/hda
```

Здесь 8193 – номер блока, полученный из вывода предыдущей команды. После этого можете смонтировать файловую систему как обычно. **КК**

11 Две головы лучше

В На работе у меня ПК с двумя мониторами и с дуальной загрузкой: Ubuntu или XP. Дома у меня тоже два монитора. Все работает замечательно (спасибо вашей статье об использовании двух мониторов), да только моя домашняя машина ужасно тормозит. На работе у меня «двухголовая» видеокарта (потому что PCIe), я установил драйверы Nvidia и получил комфортную работу с аппаратным ускорением.

Дома у меня карта AGP и карта PCI, на два источника видео. У них разные чипсеты: одна использует драйвер nvidia-legacy, другая – новую версию. Сначала я хотел установить оба набора и указать для каждой видеокарты свой драйвер в настройке *X Window*. К сожалению, оба драйвера называются nvidia, то есть мне приходится использовать официальный драйвер для одного из адаптеров вместе с открытым драйвером для другого. Не стоит и добавлять, что мой рабочий стол стал медленным и неуклюжим.

Ищу способ установить оба драйвера и как-то прописать в файле конфигурации, чтобы нужный драйвер использовался для нужного адаптера.

У меня GeForce FX 5200 (AGP) и Riva TNT2 Model 64 Pro (PCI).

Эндрю Фишер [Andrew Fisher]

О Это не разные драйверы, а разные версии одного и того же. Две различные версии модуля ядра заставить работать одновременно невозможно. Тем не менее, альтернативы имеются.

Можно, конечно, для одной из карт взять драйвер nv, но Вы и сами убедились, что он тормозит. Можно установить старую версию драйвера NVIDIA, одну из совместимых с TNT2 – подойдут 1.0.6629 или 1.0.7176, это последние версии, работающие с TNT2 и уже поддерживающие FX5200. На время это сойдет, однако с последними версиями ядер старые драйверы Nvidia имеют проблемы, так что будущее обновление ядра теоретически может все сломать. А можно поискать дешевую не-Nvidia карту для второго дисплея, или же более новую Nvidia, поддерживающую последние драйверы.

Самое же простое решение Вы использовали на работе. FX5200 – это двухголовая карта. К ней нужен лишь DVI-VGA-адаптер, если, конечно, у вас не DVI-монитор. Настройки будут те же, что и на работе. Включите в свой **xorg.conf** такие строки:

```
Section "Device"
Identifier "NVIDIA Corporation NV34 [GeForce
   FX 5200] (rev a1)-0"
VendorName "NVIDIA"
Driver "nvidia"
BusID "PCI:1:00:0"
Screen 0
```

НБ

12 Немецкая овчарка

В Я использую Linux уже более пяти лет. По роду работы мне приходится составлять много документов, вести переписку. Поэтому, когда появилась информация о *beagle* она меня сразу же очень заинтересовала. Установил FC5 с Вашего диска, KDE (в силу привычки), локаль koi8-r, попытался настроить *beagle* (**beagle-0.2.6-1.fc5.1**) и столкнулся с двумя проблемами:

1. *beagle* не индексирует файлы с русскими названиями.
2. нет поиска по файлам Microsoft Word (хотя по описанию программы вроде бы должен быть).

Вторая проблема решилась установкой внешнего фильтра

```
<filter>
  <mimetype>application/msword</mimetype>
  <extension>.doc</extension>
  <extension>.DOC</extension>
  <command>catdoc</command>
  <arguments>%s</arguments>
</filter>
```

Но проблему с файлами, названными кириллицей (системная локаль koi8-r) победить никак не удается. Можете ли вы помочь мне?

Константин Макуха

»

» **О** Проблема кроется в том, что среда времени выполнения Mono, используемая *Beagle*, ожидает увидеть на диске имена файлов в кодировке UTF-8. FAQ Mono (http://www.mono-project.com/FAQ:_Technical) рекомендует просто сменить системную кодировку, но если это по каким-то причинам нежелательно, есть обходной путь – MONO_EXTERNAL_ENCODINGS. Эта переменная окружения содержит список кодировок, в которые *Beagle* (как приложение Mono) попытается «втиснуть» найденное имя файла. Команда

```
export MONO_EXTERNAL_ENCODINGS="koi8-r"
```

добавленная перед запуском *Beagle*, решит Ваши проблемы. К сожалению, *Beagle* не предоставляет особых средств для настройки своего предстартового окружения, поэтому попробуйте добавить указанную выше команду прямо в rc-скрипт или сценарий-обертку *beagled*. Если Вы используете другие приложения Mono, которые запускаете из консоли, подумайте также о добавлении ее в свой файл **.bashrc**. **ВС**

Нагрузка на всех

В В моей фирме несколько web-серверов, используемых для Интернет/Интранет-хостинга. Мы хотим регулировать нагрузку по трафику, однако не хотим покупать балансировщик или использовать round robin DNS. Можно ли это сделать с помощью Linux?

К.Дэвидсон [K Davidson]

О Можно! GNU/Linux пользуется *Linux Virtual Server* (www.linuxvirtualserver.org), код которого, ipvs, включен в последние версии ядра. А если Ваше ядро старше 2.4.28, нужно просто взять заплатки к ним и перекомпилировать. Вы можете узнать, включен ли ipvs, таким способом:

```
cat /proc/net/ip_vs
```

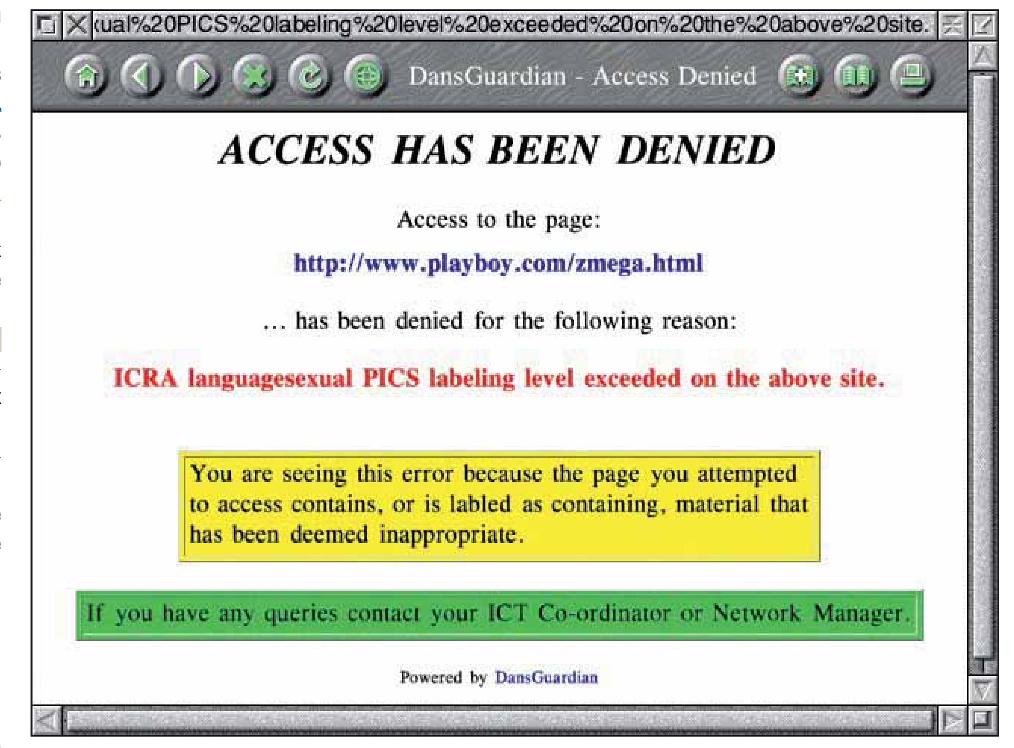
Если файл не существует, попробуйте загрузить модуль, выполнив

```
modprobe ip_vs
```

Если модуль успешно загружен или уже встроен в ядро, двигаемся дальше. На выбор есть три варианта реализации LVS: прямая маршрутизация, туннелирование или NAT. NAT настроить проще всего, но потребуется лишняя сетевая прослойка. Прямая маршрутизация работает быстрее всего, однако, она может вызвать проблемы у принимающего web-сервера.

Предположим, вы выбрали NAT. Вашему балансировщику загрузки нужны две сетевые карты: одна внутри сети, где находится web-серверы, другая в демилитаризованной зоне (DMZ) или внешней сети – в общем, там, куда посылаются ваши HTTP-запросы.

› **Наш читатель имеет два монитора, но находит работу с ними слишком медленной**



› ***DansGuardian надежно охраняет пользователя, если обновлять его почаще.***

Пусть это сеть 10.0.0.0, а сеть с web-серверами – 192.168.1.0. Присвойте машине свободные адреса, например, 10.1.0.1 и 192.168.0.1, после чего настройте таблицу маршрутизации по умолчанию для каждого web-сервера:

```
route add -net 0.0.0.0 mask 0.0.0.0 gw 192.168.1.1
```

Теперь предстоит настройка LVS: как направлять трафик по машинам. Есть несколько балансирующих алгоритмов, включая round robin, соединение с наименьшей нагрузкой и соединение с хэшированием назначения (узнать, как они работают, можно на сайте LVS). Например, выберем round-robin – при этом трафик просто отфутболивается всем серверам по очереди, прочие алгоритмы, в основном, такие же. Для операций с таблицей **Ipvs**/LVS вам понадобится программа **Ipvsadm**, вышедшая в июле 2003 г. В большинстве современных дистрибутивах она уже есть. Если ваш дистрибутив постарше, придется ее скомпилировать.

Первым делом надо настроить VIP-адреса, или виртуальные IP, где будут получать ваши запросы. Для описанной выше сети выполните следующее:

```
/sbin/ipvsadm -A -t 10.1.0.1:http -s rr
```

Теперь добавьте ваши web-серверы в VIP (подставьте свои IP-адреса):

```
/sbin/ipvsadm -a -t 10.1.0.1:http -r 192.168.1.10:http -m -w 1
/sbin/ipvsadm -a -t 10.1.0.1:http -r 192.168.1.11:http -m -w 1
/sbin/ipvsadm -a -t 10.1.0.1:http -r 192.168.1.12:http -m -w 1
```

Добавятся три web-сервера с одинаковым «весом» 1 (см. опцию -w, weight). Если вы хотите переключить на один из серверов больший трафик, просто задайте для него больший вес. Чтобы сервер не принимал трафик вообще, установите его вес в 0. **КК** LXF

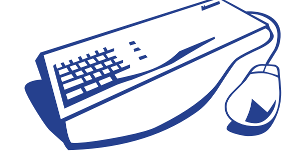
Нужна помощь!

› Для наилучшего ответа на ваш вопрос нам нужно знать как можно больше подробностей. Детально опишите конфигурацию системы. Если вы получили сообщение об ошибке, приведите текст сообщения и точно опишите вызвавшие его действия. Если у вас проблемы с оборудованием, то опишите его. Если Linux уже запущен, то выполните в root-терминале следующие команды и прикрепите к письму файл **system.txt**:

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

› Пожалуйста, помните, что сотрудники журнала НЕ являются авторами или разработчиками Linux, любых пакетов или дистрибутивов. Зачастую люди, отвечающие за приложения, выкладывают большую часть информации на web-сайты. Попробуйте почитать документацию!

Мы стараемся ответить на все вопросы. Если вы не нашли ответ на свой, это, возможно, потому, что мы уже ответили на похожий вопрос.



Лучшие новинки открытого ПО на планете

LXF HotPicks



В ЭТОМ МЕСЯЦЕ: Sonik » Glipper » Transmission » KoverArtist » Righteous » TuxFighter » HardInfo » Dolphin » Gnofract 4D » MtPaint

Звуковой редактор

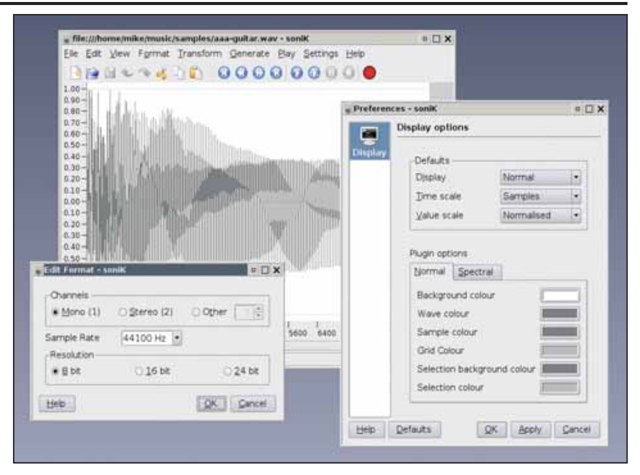
Sonik

Версия 1.0.0 **Сайт:** http://sonik.sourceforge.net

Пользователи Linux могут гордиться превосходными средствами редактирования звука: помимо отличной, основанной на *GTK* и кросс-платформенной программы *Audacity* (она позволяет вам демонстрировать друзьям развитость открытого ПО), существуют также и другие инструменты. Например, *Sonik* – основанный на KDE редактор звуковых файлов с неожиданно широким набором функций. За него следует благодарить Роба Уокера [Rob Walker], который вместо того, чтобы писать все с нуля, умело соединил в одной программе уже существующие наработки.

Поскольку *Sonik* является KDE-приложением, для его сборки вам потребуется KDE 3.3 (или новее) и пакеты Qt для разработчиков. Также для компиляции нужны библиотеки GSL и *Audiofile*, имеющиеся в большинстве дистрибутивов и, на всякий случай, на нашем DVD. Опционально вы можете использовать модули LADSPA – их наличие обеспечивает программе дополнительный функционал. Соберите программу с помощью команд **./configure**, **make** и **make install** или установите один из бинарных пакетов, благо мы и их записали на наш диск.

При запуске *Sonik* вы увидите окно мастера, который проведет вас в нужном направлении в зависимости от того, хотите ли вы записать звук, создать пустой файл или открыть уже существующий. Найдите любой WAV-файл, с которым можно поэкспериментировать. Вы обязательно что-нибудь найдете, особенно если у вас установлены какие-нибудь игры (подсказка: используйте команду **locate wav** для поиска через терминал). *Sonik* не умеет открывать файлы MP3 или OGG, поэтому вам придется сперва конвертировать их в WAV (в дальнейшем планируется добавить в программу поддержку MP3 и OGG).



› *Sonik* включает набор полезных опций, позволяя вам настроить интерфейс на свой вкус.

Когда вы откроете файл, *Sonik* отобразит волновой профиль, где можно выделять отдельные участки и проигрывать их с помощью кнопки Play на панели инструментов. Участки можно копировать, вырезать и вставлять как угодно, а функция Undo позволяет отменить все действия. В нижней части окна программы вы найдете кнопки Slider и Zoom, позволяющие перемещаться по файлу; щелчок правой кнопкой мыши по волновому профилю позволяет выбрать режим спектрограммы.

Но что же предлагает *Sonik* помимо базовых инструментов редактирования?

Вы можете регулировать качество звука (число сэмплов, каналов и разрешение), вставлять участки тишины – в конец дорожки, к примеру. Если у вас есть микрофон или соединительные кабели, то вы сможете заняться оцифровкой аналогового звука, или импортировать в программу сэмплы в форматах AIFF и SND.

Хотите выжать максимум из *Sonik*? Вам понадобятся упомянутые модули LADSPA – специальные расширения, дополнительные фильтры и эффекты. На сайте LADSPA (www.ladspa.org) есть ссылки на другие сайты с модулями, и мы советуем вам покопаться в них – вдруг вам что-то да понравится. В состав *Sonik* входят несколько полезных эффектов, например, усиление звука или волнообразные скачки громкости, однако гораздо больше интересного вы найдете в Сети. В целом, *Sonik* предоставляет все нужные функции «прямо из коробки», причем в простом и настраиваемом интерфейсе. Программа имеет все шансы стать эквивалентом *Audacity* для KDE.

Изучаем интерфейс Sonik

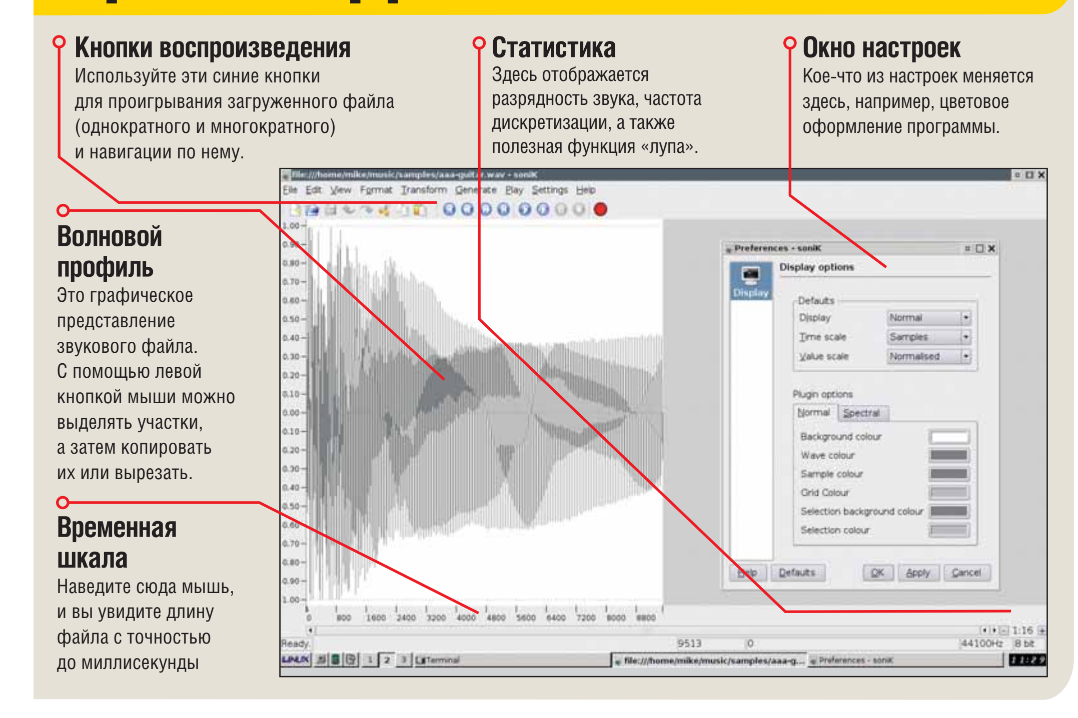
Кнопки воспроизведения Используйте эти синие кнопки для проигрывания загруженного файла (однократного и многократного) и навигации по нему.

Статистика Здесь отображается разрядность звука, частота дискретизации, а также полезная функция «лупа».

Окно настроек Кое-что из настроек меняется здесь, например, цветовое оформление программы.

Волновой профиль Это графическое представление звукового файла. С помощью левой кнопкой мыши можно выделять участки, а затем копировать их или вырезать.

Временная шкала Наведите сюда мышь, и вы увидите длину файла с точностью до миллисекунды



Менеджер буфера обмена

Glipper

Версия 0.89 **Сайт:** http://source.net/projects/glipper

Подумаешь, буфер обмена. Что может быть проще? Тем не менее, эта функция всегда приятно радует новичков в Linux. В отличие от Windows или Mac OS X, где вам нужно явно копировать текст для того, чтобы он попал в буфер обмена, система *X Window* запоминает текст, даже если он был просто выделен. Поначалу это непривычно, но при умелом использовании эта особенность может заметно ускорить вашу работу. *Glipper* – это менеджер буфера обмена, который упрощает переход на Linux путем предоставления нескольких буферов, между которыми можно переключаться.

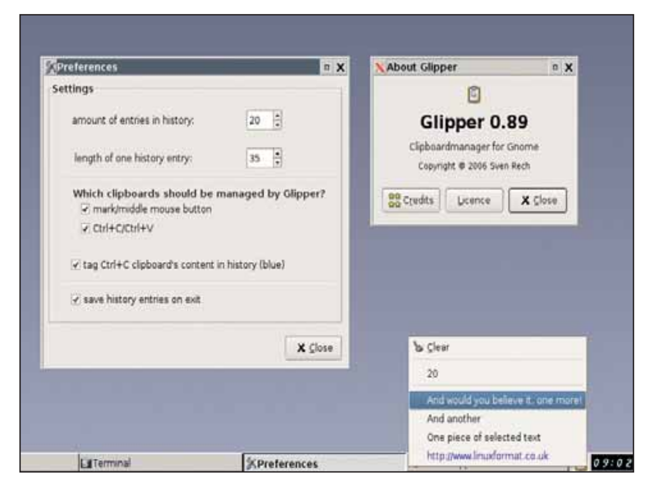
Несмотря на то, что в описании программы сказано «менеджер буфера обмена для Gnome», программа, на наше счастье, зависит только от библиотеки *GTK*, так что если вы пользуетесь другим рабочим окружением, вам не понадобится устанавливать полный комплект библиотек Gnome. Распакуйте **glipper-0.89.tar.gz**, запустите терминал и перейдите в каталог программы. Теперь введите следующие команды (последнюю – от имени суперпользователя):

```
./configure && make && make install
```

Теперь, при условии, что у вас установлены файлы разработчика *GTK*, *Glipper* будет собран и установлен в директорию **/usr/local/bin**. После этого вы можете дать команду **glipper** от имени обычного пользователя, и программа запустится.

Не удивляйтесь, пытаясь найти запущенную программу. У нее нет основного окна, и она не показывает каких-либо сообщений. Вместо этого в системном лотке появится маленькая иконка (в *IceWM* и *Xfce* она будет на основной панели). Щелкнув по ней, вы увидите меню, которое отображает выделенный в данный момент текст. Меню обновляется в реальном времени – вы можете проверить это, выделяя различные кусочки текста и щелкая по иконке программы.

«Glipper помещает небольшую иконку в системный лоток вашего оконного менеджера.»



› **Здесь меню *Glipper* отображает три фрагмента, которые мы скопировали в буфер обмена.**

Предположим, что вы работаете с каким-либо кодом и копируете строку в буфер обмена. Когда вы скопируете другую строку, первая переместится в *Glipper* на позицию ниже, однако вы можете сделать ее снова текущим содержанием буфера, щелкнув по ней в меню. Щелчок правой кнопкой мыши вызывает окно настроек, где можно задать количество запоминаемых элементов, а также способ копирования (при явном нажатии Ctrl+C, как Windows, или просто при выделении – в стиле X11?). Да, *Glipper* чрезвычайно полезен на рабочем столе, причем его лучшими свойствами можно считать незаметность и низкое потребление памяти.

Клиент BitTorrent

Transmission

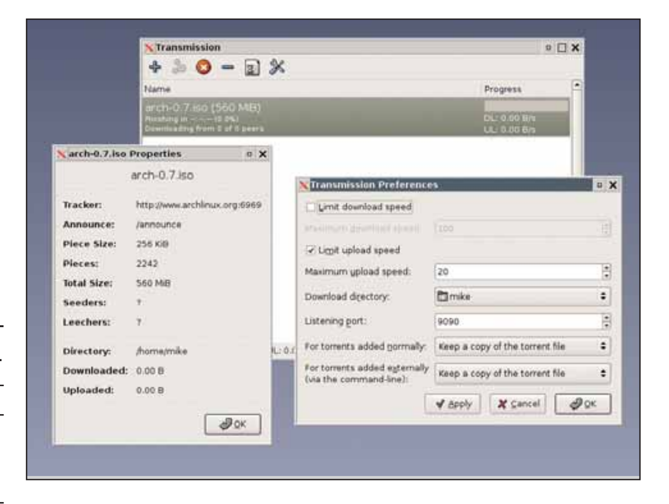
Версия 0.6.1 **Сайт:** http://transmission.m0k.org

Приятно видеть, что *BitTorrent* стал гораздо большим, чем просто приятная безделушка, и многие люди используют его для распределения трафика при скачивании ISO-образов дистрибутивов Linux. С одной стороны, это способствует нелегальному копированию ПО и музыки, но с другой стороны, это идеальный способ загрузки, ведь центральный сервер практически не нагружается. Для Linux существует достаточно клиентов *BitTorrent*, однако большинство из них слишком сложны и потребляют много ресурсов (привет тебе, *Azureus*). С *Transmission* вам будет гораздо легче работать, так как эта программа не перегружена лишними функциями.

Установка *Transmission* ничем не отличается от таковой в *Glipper*. Программа состоит из базовой части, работающей в командном режиме, и графического интерфейса. Не забудьте, что для сборки последнего вам понадобятся файлы разработчика *GTK*. На этом зависимости заканчиваются, поэтому скрестите пальцы и начинайте сборку. Если *Transmission* покажется вам самым лучшим решением для работы с *BitTorrent* и вы захотите поделиться ею с друзьями, вы будете приятно удивлены наличием версии для Mac OS (и даже BeOS, если у вас есть особо чокнутые друзья!).

Собрав приложение, наберите **Transmission-gtk**, чтобы запустить графическую оболочку. Теперь можно вздохнуть с облегчением: программа исключительно проста в использовании. Щелкните по кнопке Add (Добавить, +) и выберите заранее скачанный файл **.torrent**. Все остальное *Transmission* сделает сама. В строке состояния появится информация о закачке, включая пропускную способность соединения и расчетное время. В любой момент вы можете щелкнуть правой кнопкой мыши на панели и получить дополнительные сведения о torrent-закачках, например, число seeder'ов (предлагающих вам скачать файл у них) и число активных запросов на этот файл. Эти сведения дают представление о том, насколько устойчивым является ваше соединение.

«С Transmission легко управиться; в нем нет лишних функций и подводных камней.»



› **Интерфейс *Transmission* сделан предельно простым для того, чтобы быстро выполнять свою работу.**

Программа включает несложные настройки в диалоге Options, где вы можете установить ограничение на скорость закачки – это очень полезно, если во время скачивания вы хотите заниматься другими делами в Интернете, пока *BitTorrent*-клиент наблюдает за портом и выделяет на вашем диске место под закачку. В программе пока нет справочного руководства, однако если вы когда-либо скачивали torrent-файлы, вы мгновенно разберетесь, что к чему. Если вам нужна простая закачка, без графиков и бесконечной статистики, которой бомбят вас другие программы, то *Transmission* должна вам понравиться.

Программа для создания обложек к CD/DVD

KoverArtist

Версия 0.4.1 **Сайт:** www.kde-apps.org/content/show.php?content=38195

На полках редакции LXF валяется бесчисленное множество кое-как подписанных дисков. Обычно это загрузочные диски дистрибутивов, которые мы тестируем, но иногда попадаются и резервные копии, заслуживающие лучшей участи, чем лежать «голышом» в стопке. Если вы когда-либо записывали музыкальные альбомы, фильмы или делали архивные копии своих любимых дисков, то вам наверняка хотелось, чтобы они не только хранили важные данные, но и выглядели достойно. В этом вам может помочь *KoverArtist*.

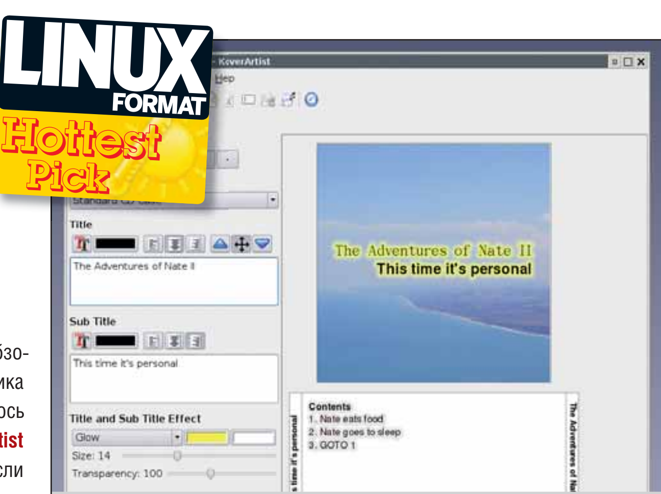
KDE-приложение для создания обложек для дисков было написано Стефаном Теффернером [Stefan Taferner] из-за необходимости промаркировать его коллекцию фильмов. Однако программа достаточно гибка для создания самых разных типов обложек – фильмов, аудио дисков, резервных копий на DVD и даже обложек для самодельных дистрибутивов. Да, если вы только что закончили работу над крутым линуксовым Live CD и хотите придать ему солидный вид, это как раз то приложение, которое вам нужно. Забудьте о прозрачных коробках и процарапанных ручкой надписях на тыльной стороне диска.

Разработчики *KoverArtist* выпустили универсальный RPM-пакет (**KoverArtist-0.4-1.i586.rpm**), который вы, скорее всего, сможете установить на вашу систему. Пользователи Debian и Ubuntu могут попробовать конвертировать его в DEB-пакет при помощи Alien, однако если это не поможет, попытайтесь собрать программу из исходных текстов (принцип описан в обзоре *Glipper*), не забыв про пакеты разработчика KDE и Qt. Если все собралось и установилось правильно, просто введите команду **koverartist** в окне терминала для запуска программы (если не получается, попробуйте указать полный путь к исполняемому файлу: **/usr/local/kde/bin/KoverArtist**) [а еще лучше – замените при сборке вызов **configure** на **configure --prefix=`kde-config --prefix`**, – прим.ред.].

Выбор обложки

За дизайн *KoverArtist* Стефан Теффернер заслуживает награды: гибкость в создании обложек разных размеров никак не противоречит разумной структуре интерфейса. В окне программы есть три вкладки: Project, Options и 1 (номер диска, в начале здесь всегда стоит единица), в каждой из которых есть простые для понимания поля ввода. Во вкладке Project вы задаете тип обложки – это может быть обычный футляр для CD, слим-бокс или коробка под DVD. Ваш выбор автоматически влияет на размер рабочей области и править вручную ничего не нужно.

Затем вы вписываете заголовок и подзаголовок диска; по умолчанию эти надписи аккуратно подсвечиваются и располагаются по центру обложки (эффекты можно менять). Переключившись на вкладку Options, вы можете задавать изображения для передней и задней обложки. Для этого вы можете заранее отсканировать нужные изображения или же подготовить их в *Gimp* и лишь потом вставить их в проект. Наконец, следует оформить тыльную сторону футляра, куда можно поместить оглавление содержимого диска – нумерацию песен аудио диска, раскадровку фильма и даже список сессий обычного диска. Выбирать действительно есть из чего, и вам не обязательно помещать сюда просто список файлов – можно, например, поместить инструкции по применению Live CD.

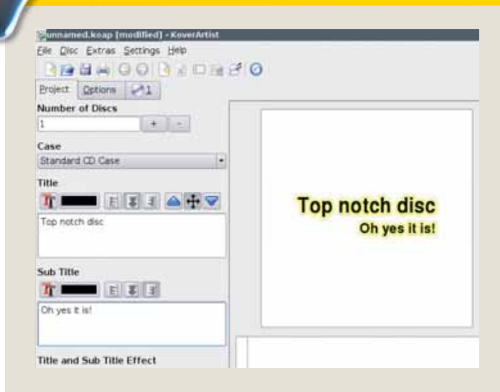


› Справа вы видите переднюю и заднюю обложки на стадии оформления

Итак, в этой программе действительно есть все для создания красивых и информативных обложек для дисков. Но в *KoverArtist* есть и некоторые дополнительные возможности, например, импорт сведений о музыкальных альбомах из базы данных FreeDB и импорт файлов из *KMediaFactory*. Готовые обложки можно сохранить для дальнейшей распечатки или редактирования. Файлы обложек хранятся в формате XML, что позволяет вам при желании изменять их вручную.

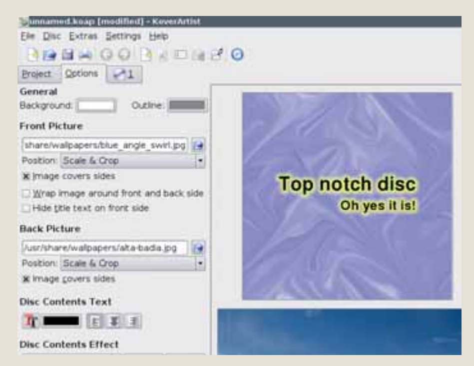
В *KoverArtist* все находится на своих местах: программой очень легко пользоваться, можно создавать сложные макеты обложек и вообще, делать что угодно без помех. Программа в действительности гораздо ближе к версии 1.0, чем предполагает номер версии, поэтому вам стоит попробовать ее, если вы хотите сделать свои диски презентабельнее.

Шаг за шагом: Создание обложки



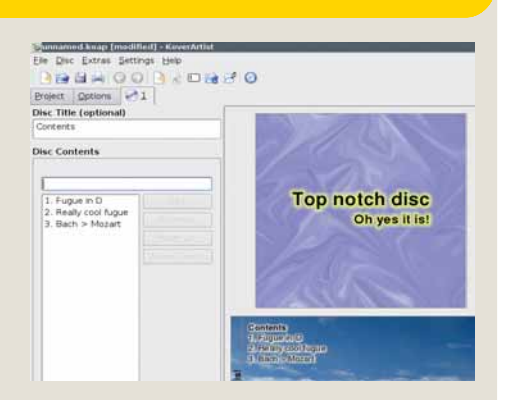
» Выберите размер

Установите размер вашего футляра, выбрав его из списка. Затем впишите заголовок и подзаголовок и выберите подходящий шрифт.



» Установите картинку

Щелкните по вкладке Options и выберите изображение или фон для лицевой и тыльной сторон футляра.



» Добавьте оглавление

Выберите вкладку с номером диска, задайте описание содержимого диска и, наконец, щелкните по кнопке Print (Печать)!

HotGames Развлекательные приложения

Логическая игра

Righteous

Версия 0.6 **Сайт:** http://atomsknінja.googlepages.com/righteous!

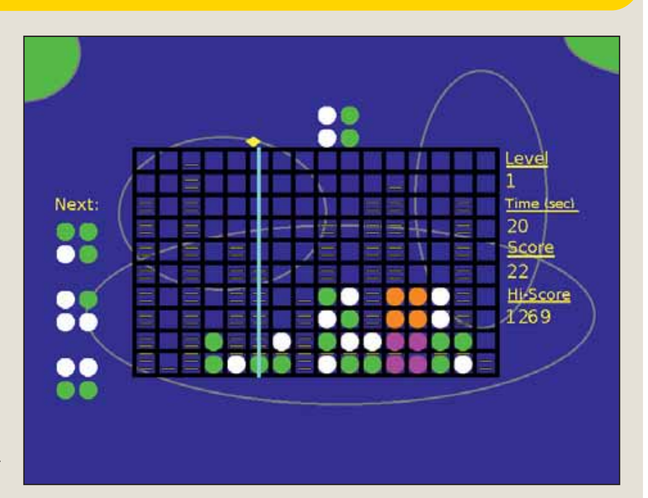
Карьера Тетсуя Мизугучи [Tetsuya Mizuguchi] в создании игр очень интересна. Наиболее известны две из его разработок: игры *Sega Rally* и *Lumines*. Конечно, мы мечтаем увидеть открытую игру на основе первой, но, честно говоря, это звучит слишком амбициозно. Тем не менее, у нас теперь есть открытый клон второй. *Righteous* от AtomskNinja Software Бена Халла [Ben Hull], представляет собой блочную игру, в которой сочетаются Tetris, *Columns* и другие игры. Оформление здесь не такое сногсшибательное, как в *Lumines*, но зато сохранено самое главное – игровой процесс.

Для того, чтобы запустить игру, распакуйте файл **Righteous-0.6.tar.gz**, перейдите в директорию игры и дайте команду **build.sh**. Если у вас установлены пакеты разработчика SDL, то эта команда скомпилирует главный исполняемый файл (*Righteous*), который можно будет сразу запустить. Для того, чтобы понять суть игры, запустите демо-сценарий: все работает быстро, но можно уловить стратегию.

Смысл игры в *Righteous* состоит в выравнивании цветных шариков, которые падают сверху в виде блоков 4x4. Их можно вращать и передвигать вправо или влево, составляя внизу блоки одного цвета. Играть в игру так же легко, как и в остальные блочно-мозаичные игры, но несмотря на это, уже совсем скоро вы подбираетесь к верху экрана и начинается настоящий кошмар! Когда блокам некуда падать, игра заканчивается.

Составленные блоки одного цвета не исчезают сразу. Вместо этого через экран время от времени проходит «линия времени», которая «подметает» готовые блоки. Во время игры звучит кое-как сделанная скучная музыка; кроме этого, вы вряд ли будете в восторге от графики, которая здесь явно не на высоте, однако это все не важно – важен процесс! Эта игра удачно сочетает достоинства разных блочных игр.

Проект AtomskNinja планирует сделать из *Righteous* нечто большее, чем клон *Lumines* и добавить новые возможности и сюжеты. Это правильный путь, особенно с учетом того, что обычно разработчики берут один и тот же сюжет и добавляют свои детали только для того, чтобы быть «не такими как все».



› Оцените трилинейную интерполяцию текстур! Или не оцените.

Action

TuxFighter

Версия 0.52 **Сайт:** http://pygamebook.sourceforge.net

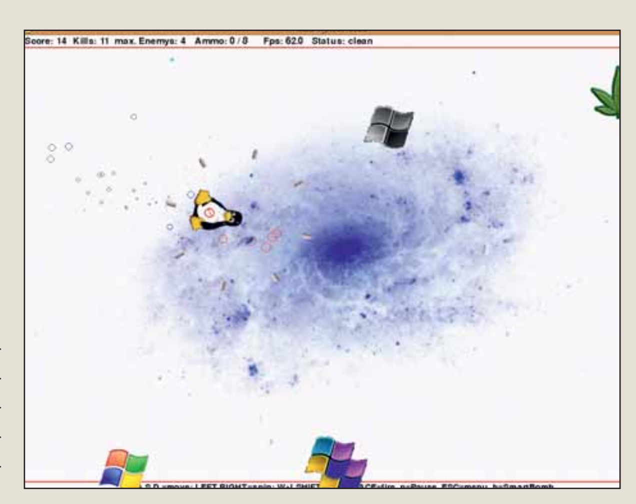
Мы всегда представляли себе Тукса – талисмана Linux – в образе жизнерадостного баловня-пингвина, которому все побоку. Разработчики *TuxFighter* так не считают, поэтому они создали игру, в которой пингвин отражает натиск логотипов Windows. Тукс воюет с Microsoft, но вовсе не на рынке ОС. Вместо этого нашего пингвина заслали в космос, где он занимается ракетными обстрелами. Игра похожа на Asteroids, только посвящена операционным системам.

Разработчики *TuxFighter* подготовили deb-пакет, который работает на Ubuntu Dapper (6.06) и должен также установиться на современной версии Debian. Если же вы работаете на RPM-дистрибутиве, то можете установить игру из архива **TuxFighter52.tgz**, при условии, что у вас есть Python и пакет привязок *PyGame*.

После запуска игры вы сразу же увидите меню, где можно переназначить клавиши или изменить сложность игры (враги вас атакуют сразу или просто лениво снуют по экрану). Начните игру, выбрав пункт Start Single Player Game (одиночная игра), и принимайтесь за мышь с клавиатурой: стрелками влево и вправо вы поворачиваете пингвина, пробелом стреляете ракетой, а клавишами W, A, S и D передвигаетесь в пространстве. Тем, кто хоть раз играл в стрелялки, эти настройки покажутся знакомыми.

Цель игры состоит в уничтожении логотипов Windows вашими ракетами. Если, летая по экрану, вы заденете врага, до погибните (в игре, к сожалению, только одна жизнь!). Казалось бы, вполне обычные *Asteroids*, но некоторое разнообразие в игру вносят значки дружественных дистрибутивов. Завидев логотип дистрибутива, подлетайте к нему и тогда и пингвина появятся дополнительные способности. Значок Ubuntu добавляет защиты, а *Wine* – замораживает логотипы Windows, позволяя вам избежать трудных ситуаций.

Кому-то эта игра может показаться простой бомбежкой Windows, но мы склонны видеть в ней большой потенциал. Было бы здорово включить в игру дополнительные значки, например *OpenOffice.org*, который бы притормаживал врагов, или самого Херда (Hurd), появление которого означало бы, что теперь вы точно никогда не выиграете... Все же, игра проста и также притягательна, как и *Asteroids*, только с линуксовским сюжетом. Нам понравилось!



› Талисман Linux всегда представлялся нам миролюбивым созданием – но и он поддался темной стороне Силы.



Системный анализатор

HardInfo

Версия 0.4.1 **Сайт:** http://hardinfo.berlios.de

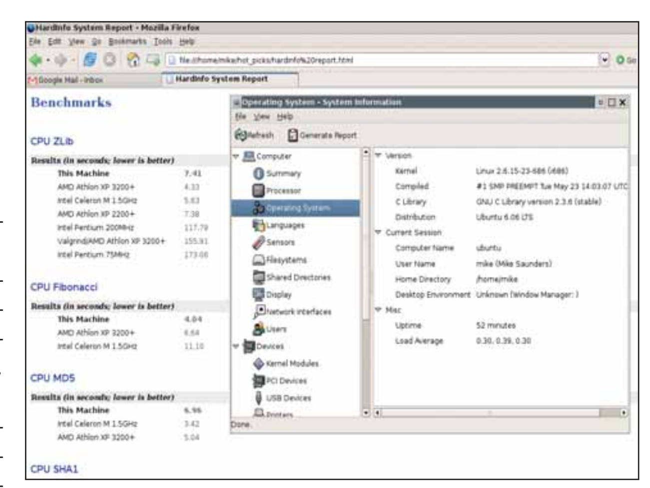
Приведем в пример ситуацию, с которой вы наверняка сталкивались вне зависимости от того, как долго вы работаете в Linux. На интернет-форуме вы пытаетесь описать проблему с дистрибутивом, и кто-то просит подробнее рассказать о вашей системе. Понятно, что вам хотят помочь, но дать точный совет не так-то просто! Вам придется углубиться в командную строку и выяснить точные версии ядра, библиотеки Glibc, X-сервера и много чего еще, не считая точных названий комплектующих вашего ПК. *HardInfo* пытается исправить эту ситуацию, отображая все нужны сведения в виде четкой и логичной структуры.

Скажем отдельное спасибо разработчикам *HardInfo*, которые сделали для своей программы Автопакет. Это означает, что вы можете забыть про зависимости и сборку исходных текстов. Скопируйте **HardInfo-0.4.1.x86.package** с нашего DVD на свой диск, сделайте его исполняемым (в свойствах файла) и дважды щелкните по нему для установки. Можно сделать и по-другому: откройте терминал в директории, где лежит файл и дайте команду: **sh HardInfo-0.4.1.x86.package**. После этого можно запустить программу командой **HardInfo**.

Несмотря на свое название, *HardInfo* собирает информацию не только о вашем «железе», но и о системе в целом, включая версию ядра и подробности настроек X-сервера. Наиболее важные данные попадают на экран Summary (Сводка), но этим дело не ограничивается, и вы можете ожидать самых подробных сведений практически обо всем: директории с совместным доступом, языки системы, USB-устройства, съемные диски, мониторы оборудования и многое другое описаны с большой подробностью. Обзор оборудования во многом напоминает (мало)известный Диспетчер устройств в Windows.

На сегодняшний день *HardInfo* – это лучшая утилита для анализа состава вашей системы, но на этом ее возможности не заканчиваются. В составе программы вы найдете средства замера производительности, которые тестируют ваш CPU с самых разных сторон, после чего позволяют сравнить результаты с замерами для других CPU. Так вы можете узнать свое место относительно Pentium 75 или Athlon XP 3200 и при наличии неисправности быстро ее обнаружить.

Но что еще лучше, *HardInfo* умеет оформлять любые отчеты в виде HTML-страниц, которые вы можете закачать на интернет-сервер. Так вы получите больше откликов на призыв о помощи от других пользователей Linux. Эту программу просто необходимо установить!



› *HardInfo* отображает детальные сведения об ОС, в то время как на заднем плане виден аккуратный HTML-отчет.

«HardInfo — это больше, чем информация об оборудовании.»

Файловый менеджер

Dolphin

Версия 0.5.2 **Сайт:** http://enzosworld.gmxhome.de

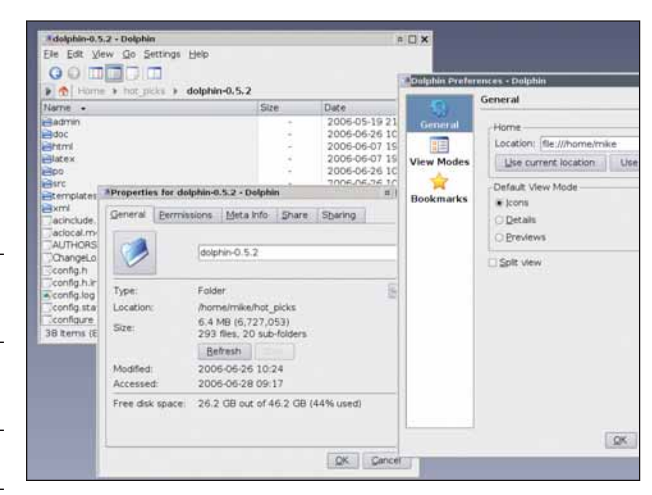
Konqueror содержит столько всевозможных функций, что в какой-то момент его стали ругать за перегруженность и чрезмерную сложность – это не только файловый менеджер, но и web-браузер, просмотрщик документов и многое другое. Петер Пенц (Peter Penz), автор *Dolphin*, сосредоточил свои усилия исключительно на управлении файлами и не стал искушать пользователей дополнительными, не имеющими к файлам прямого отношения, функциями. Он специально подчеркивает, что его программа не конкурирует с *Konqueror*, а «оптимизирует интерфейс для работы с файлами». Сейчас программа имеет статус «альфа», в то время как финальный релиз 1.0 должен выйти в сентябре.

Для того, чтобы собрать программу из исходных текстов, вам понадобятся файлы разработчика KDE и *Qt*, установив которые, вы сможете скомпилировать программу обычным способом. Для запуска программы введите в терминале команду **Dolphin**. По умолчанию, главное окно программы представляет собой одну панель с расположенными «в сетке» иконками, однако виды можно переключать с помощью панели инструментов в верхней части окна – имеются также детализированный список и двупанельный режим. Под этими инструментами расположена область навигации, которая помогает перемещаться между папками – при желании эту область можно превратить в привычную строку адреса и вводить путь к директории вручную.

Программа позволяет выполнять все базовые действия над файлами (копирование, перемещение, переименование, удаление), а также некоторые дополнительные, в число которых входит сжатие папок и возможность открытия файла в разных приложениях.

Наиболее продвинутой функцией *Dolphin* является просмотр свойств файла или папки, во время которого вам предоставляется самая подробная информация: занимаемое на диске место, число поддиректорий (если есть), время доступа и права. Отдельное спасибо разработчику за внятное объяснение прав файлов и каталогов, которые даются в виде понятных человеку описаний, вроде такого: «Владелец может просматривать и изменять содержимое». Это явно удобнее сложной системы «флажков» в классическом стиле Unix.

Благодаря подсистеме KIOSlave, встроенной в KDE, *Dolphin* можно совершенно прозрачно работать с сетевыми каталогами через FTP, SSH и другие протоколы. Программа также поддерживает раннюю реализацию системы обмена файлами *Avahi* – пока еще здесь не все гладко, но мы надеемся, что эта функциональность будет доработана к моменту выхода *Dolphin 1.0*. На данный же момент *Dolphin* составляет хорошую компанию *Konqueror* и идеально подходит тем, кто хочет сосредоточиться на управлении файлами.



› Dolphin отображает файлы в режиме списка. Иконки можно сделать больше – на ваше усмотрение.

Генератор фрактальных изображений

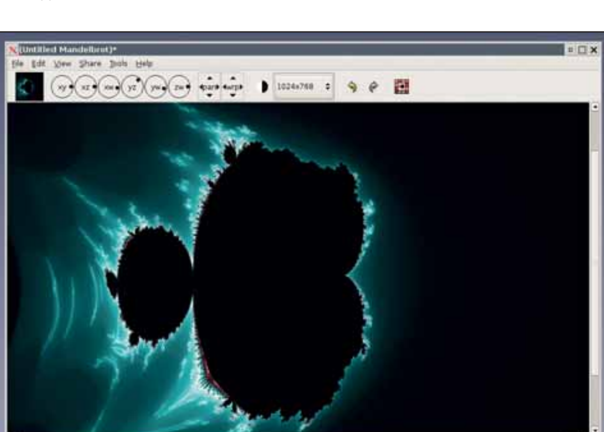
Gnofract 4D

Версия 3.0 **Сайт:** http://gnofract4d.sourceforge.net

Чуть больше двух лет назад в **LXF53** мы рассматривали *Gnofract 1.9*, который умел делать больше, чем просто отрисовывать фрактальные изображения. Программа работала с фракталами, как с «разными сторонами одного 4-мерного фрактального объекта» – сумасшедший уровень математики, который приносил роскошный результат. Вам предоставляется необъятный контроль над изображением, включая возможность отдалять или приближать объект по своему желанию. Распакуйте **gnofract4d-3.0.tar.gz**, перейдите в получившуюся директорию и дайте сначала команду **./setup.py build**, а затем **./gnofract4d** для запуска программы.

Изменения, произошедшие со времен версии 1.9 включают новые возможности публикации изображений – теперь их можно закачать на Flickr или отправить по E-mail, не покидая программы. Также появилась опция Render to Disk (Отрисовать на диске), позволяющая создавать файлы очень большого размера, и Warp Param, расширяющая список фрактальных формул. Появился редактор градиентов, который умеет импортировать градиентные заливки из *Gimp*.

Gnofract 4D кажется сложным приложением, к тому же большая часть формул нам мало что говорит. Но несмотря на это, все отлично работает, даже если вы просто хотите создать какое-нибудь фрактальное изображение. В действительности, вам совсем не обязательно знать о том, что происходит за отрисовкой сцен. Нет более легкого способа создать уникальные обои для рабочего стола, чем сделать это в *Gnofract 4D*.



› Стандартный фрактал Мандельброта повернут по осям Y и Z, что создает приятный размытый эффект.

Редактор изображений

MtPaint

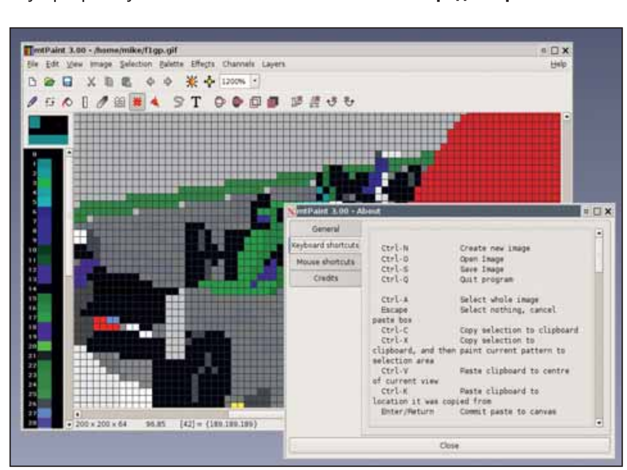


Версия 3.00 **Сайт:** http://mtpaint.sourceforge.net

С того времени, когда мы рассматривали *MtPaint* версии 0.45 в **LXF63**, этот редактор успел сильно повзрослеть. Тогда *MtPaint* годился в качестве Linux-аналога *MS Paint* – приложения для простого рисования и редактирования, но не для серьезной работы. Теперь его разработкой занимается Дмитрий Грошев, и в этой связи за последние два года произошли важные изменения.

Наиболее заметные из них таковы: инструменты смазывания изображения и правки текста; захват экрана, поддержка прозрачности в форматах PNG и GIF; поддержка drag-n-drop, инструмент выделения «лассо» и три новых локализации программы. Web-дизайнеры порадуются поддержке анимированных GIF-файлов, а художники оценят то, что теперь программа распознает силу нажатия при работе с планшетами. Как и прежде, вы можете скомпилировать *MtPaint* с поддержкой как *GTK2*, так и *GTK1*, что добавляет разработчикам много лишней работы, но зато позволяет собрать программу на старых дистрибутивах.

MtPaint проделал большой путь от простой программы до вполне продвинутого графического редактора. Он не может сравниться с *Gimp* в плане функциональности, но, несмотря на это, содержит все необходимое для повседневного использования. Интерфейс покажется привычным для всех, кто хоть раз видел *MS Paint*, поэтому, если вам нужно подправить изображения, но вы боитесь потонуть в море функций *Gimp*, обратите внимание на эту программу. **LXF**

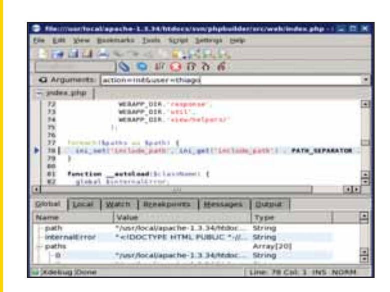


› При увеличении *MtPaint* рисует сетку для более удобного попиксельного редактирования.

Также выпущены

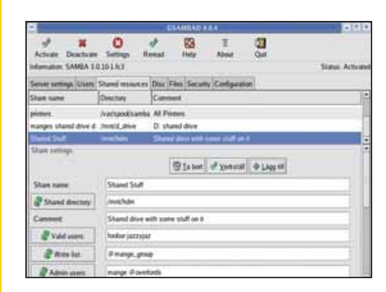
Новые и обновленные программы, заслуживающие вашего внимания...

- **Arena 0.9.5** С-подобный язык скриптов **www.minimalinux.org/arena**
- **PoDoFoBrowser 0.2** Браузер структуры PDF-файлов **http://podofo.sourceforge.net**
- **QComicBook 0.2.8** Просмотрщик сетевых комиксов **http://linux.bydg.org/~yogin**
- **ClamTk 2.20** Интерфейс к антивирусу *ClamAV* **http://clamtk.sourceforge.net**
- **Protoeditor 1.0** Текстовой редактор для KDE **http://protoeditor.sourceforge.net**



› Protoeditor: баг не пройдет.

- **HomeBank 3.2a1** Менеджер финансов **http://homebank.free.fr**
- **Flail 0.2.2** Консольный почтовый клиент на Perll **http://flail.org**
- **Inkscape 0.44** Мощный редактор векторной графики **www.inkscape.org**
- **Pen 0.17.1** Балансировщик нагрузки на HTTP и SMTP **http://siag.nu/pen**
- **Rawstudio 0.2** Основанный на *GTK* конвертер RAW-фотографий **http://rawstudio.org**
- **Morla 0.9** Редактор документов RDF **www2.autistici.org/bakunin/morla**
- **GSAMBAD** Административный интерфейс для Samba, основанный на *GTK* **www.gadmintools.org**



› GSAMBAD: импортируй и разделяй!

- **FinScribe 0.6** Программа для проверки web-страниц **http://hww3.riverweb.com/space/pike/FinScribe**
- **KGliv 0.5.1** Просмотрщик графических файлов с использованием OpenGL **www.kde-apps.org/content/show.php?content=40392**
- **sshproxy 0.4** Прокси-сервер на *Python* **http://penguin.fr/sshproxy**
- **Mesk 0.1.2** Музыкальный проигрыватель, написанный на PyGTK **http://mesk.nicfit.net**

LXF DVD 84

Не один, не два, а целых три дистрибутива в этом месяце (и BSD)!



Майк Сондерс любовно подбирает содержимое диска Linux Format, а также поддерживает сайт www.linuxformat.co.uk

Что там у вас в дисководе?

Добро пожаловать на обновленные страницы, посвященные диску LXF! Самое большое новшество – раздел HotPicks, переехавший к нам сюда со своего прежнего места. Нам кажется, в этом куда больше смысла: ведь приложения, которые рассматриваются в нем, всегда есть и на нашем диске, так почему бы не объединить все, что касается содержимого диска, для большего удобства? Вдобавок HotPicks стали полезнее, теперь здесь дается объяснение большинства новых возможностей. И, как всегда, хотелось бы узнать ваши впечатления и предложения.

Перейдем к содержанию DVD этого месяца. Нам удалось втиснуть на диск пять огромных операционных систем; две из них – самые большие – загружаются прямо с DVD. И это – отличные ОС: Knoppix и SimplyMepis, два самых популярных Live-дистрибутива, пользующиеся широкой известностью; их можно установить на ваш жесткий диск. Ну, а если вам по душе осваивать нечто более эзотерическое, взгляните на PC-BSD, новую разновидность FreeBSD, с ориентацией на рабочий стол и быстро растущей армией поклонников.

А уж если и это кажется вам привычным руслом, попробуйте Minix 3, новый релиз Unix-подобной операционной системы, благодаря которой Линус Торвальдс [Linus Torvalds] начал разработку ядра. Ах да, для старых машин у нас есть дистрибутив Zenwalk с Xfce. Вас ждет также обычная подборка новых релизов программ и основных компонентов системы – так что открывайте дисковод, вставляйте диск, и вперед!

mike.saunders@futurenet.co.uk

Краткое содержание DVD

ЖУРНАЛ
3D игры .... Файлы из учебника Ogre
Greasemonkey ..... Код из учебника
OOo Basic .... Дополнительный код из учебника Марка Бейна
Roundup .... Открытые утилиты доступа к удаленному рабочему столу

РАБОЧИЙ СТОЛ
Electric Sheep ...... Забавный хранитель экрана
GCstar .... Менеджер коллекций
Karmen .... Оконный менеджер
ScreenKast .... Программа для записи с экрана
WotD .... Создатель словарей

РАЗРАБОТКА
Disassembler .... Дизассемблер исполняемых файлов
Gambas .... среда разработки Basic
Mono .... Среда разработки NET

ДИСТРИБУТИВЫ
Knoppix .... Популярный Live-дистрибутив
SimplyMepis .... Live дистрибутив на базе Ubuntu
Minix3 .... Операционная система
PC-BSD .... Настольная версия FreeBSD
Zenwalk .... Скоростной дистрибутив Linux
MOPS .... Дистрибутив на базе Slackware (сторона 2)

ИГРЫ
Gnome Hearts ...... Карточная игра
Sturmbahnfahrer .. Гонки
Tavli 3D .... Греческие нарды

ГРАФИКА
DigiKam .... Менеджер фотографий
Lives .... Редактор видео
RPhoto .... Обработка изображений

HOTPICKS
Dolphin .... Файловый менеджер
Glipper .... Буфер обмена
Gnofract 4D .... Генератор фрактальных изображений
HardInfo .... Анализ системы
KoverArtist .... Редактор обложек для CD/DVD
MtPaint .... Редактор изображений
Righteous .... Головоломка
Sonik .... Звуковой редактор
Transmission .... Клиент BitTorrent
TuxFighter .... Игра в жанре action

INTERNET
Firefox .... Web-браузер
GProFTPD .... Инструмент администрирования ProFTPD
Opera .... Интернет-пакет

ОФИС
Eqonomize .... Финансовый менеджер KDE
GnuCash .... Финансовый пакет
OpenOffice org ..... Офисный пакет

ЗВУК
Beast .... Инструмент создания музыки
BMPx .... Компактный медиа-плейер
Jokosher .... Многоканальная студия

СИСТЕМА
John the Ripper ... Поиск слабых паролей
KLogWatch .... Сетевой монитор
Qemu .... Эмулятор ПК
QRepoClean .... Программа очистки кэша
Xen .... Виртуальный монитор

ГЛАВНОЕ
Avifile .... Библиотека для считывания/записи AVI файлов
CheckInstall .... Программа создания бинарных пакетов
Coreutils .... Утилиты командной строки
GLib .... Низкоуровневая библиотека C
Glibc .... Библиотека GNU C
GTK .... Набор инструментов пользовательского интерфейса.
Jigdo .... Создатель ISO-образов.
Kernel .... Свежая версия ядра Linux.
libsigc .... Система обратных вызовов для C++.
libXML .... Анализатор и набор инструментов XML.
Ncurses .... Инструментарий текстового режима.
Python .... Язык программирования.
Rawrite .... Программа записи образов на дискеты.
SBM .... Smart Boot Manager.
SDL .... Мультимедиа-библиотека.



Информация о диске

Внимательно прочтите это перед тем, как использовать DVD-диск.

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;
имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;
имя_программы-1.0.1.tar.gz – обычно это исходный код;
имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;
имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;
имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

KNOPPIX

Быстрый, милый – пять баллов!
Самый популярный живой дистрибутив

- Ядро 2.6.18, KDE 3.5.2
- Установка на жесткий диск
- X.org 7.0, Firefox 1.5.0.3
- Запись на разделы NTFS

также на диске

- PC-BSD 1.2: Надежная ОС для рабочего стола
- Zenwalk Linux 3.0: Быстрый дистрибутив на основе Xfce
- Minix 3: Эта ОС вдохновляла Линуса Торвальдса
- Opera 9.02: Браузер, почта, обмен файлами
- OpenOffice.org: открытый офисный пакет
- Игры: гонки и головоломки



» Дистрибутив для дома и офиса
Стартует прямо с DVD, с возможностью установки на жесткий диск
KDE 3.5.3 - OpenOffice.org 2.0.2 - Основан на Ubuntu





LINUX FORMAT 4 GIGABYTE DVD DUAL BOOTING DVD / 4 DISTRO





Содержание LINUX FORMAT DVD Октябрь 2006

Сторона 1

Рабочий стол
Electric Sheep – хранитель экрана
GCStar – программа для управления коллекциями
Karmen – легкий оконный менеджер
ScreenKast – программа для записи событий на экране
WotD – апплет «слово дня»

Дистрибутивы
Knoppix – популярнейший в мире LiveCD
Mepis – дистрибутив на базе Ubuntu
Minix3 – ОС с микроядром
PC-BSD – переработанная для пользовательских нужд FreeBSD
Zenwalk – легкий дистрибутив на базе Slackware

Разработка
Disassembler – графический дизассемблер для Linux
Gambas – среда разработки для Gambas Basic
Mono – открытая реализация .NET

Игры
Gnome-Hearts - «червы» для Gnome
Sturmbahnfahrer – автосимулятор с реалистичной физикой
Tavli3d – греческая версия нард

Графика
DigiKam – менеджер фотоколлекций
LiVES – продвинутый редактор видео
Rphoto – программа для просмотра фотографий

Звук
Beast – модульный синтезатор
BMPx – компактный аудиоплеер с менеджером коллекции
Jokosher – мощный аудиоредактор с поддержкой множества дорожек

Графика
Горячие новинки
Dolphin - файловый менеджер для среды KDE
Glipper - менеджер буфера обмена для среды GNOME
Gnofract4D - программа-генератор фракталов
HardInfo - утилита для получения информации об оборудовании
KoverArtist - программа для быстрого создания обложек к компакт-дискам.
MtPaint - графический редактор специально для пиксель-арта.
Righteous - популярный паззл
Sonik - открытый цифровой аудиоредактор для KDE.
Transmission - легкий и быстрый BitTorrent-клиент.
TuxFighter - аркадная игра наподобие Asteroids.

Интернет
Firefox – свободный веб-браузер
GproFTPD - быстрая и простая в использовании утилита для настройки Proftpd-сервера
Opera – самый быстрый веб-браузер

Офис
Eqonomize! - приложение для управления финансами
GnuCash – продвинутый менеджер персональных финансов
OpenOffice.org – открытый офисный пакет

Система
John The Ripper – программа для вскрытия паролей Unix
KlogWatch – программа для слежения за логами
QEMU – открытый эмулятор процессора
QrepoClean – скрипт для очистки кэш-папок от старых пакетов
Xen – открытый виртуализатор

Поставляется вместе с журналом LINUXFORMAT номер 10(83) Октябрь 2006

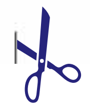
Сторона 2

MOPS Linux 5.0 – отечественный дистрибутив на базе Slackware 11

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен на Уральском электронном заводе., 620066, Россия, г. Екатеринбург, ул. Комвузовская 17-203, Лицензия МПТР России ВАФ № 77-15



Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF

Дистрибутив Linux

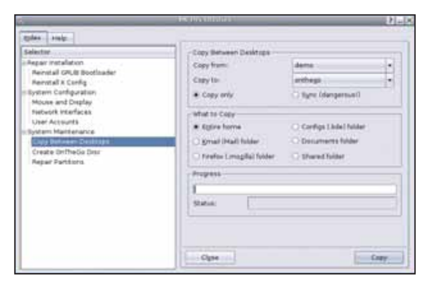
SimplyMepis 6.0

SimplyMepis, бесспорно, один из самых замечательных дистрибутивов для рабочего стола: он невероятно прост в установке и использовании, включает современные программные пакеты и обеспечен поддержкой весьма процветающего сообщества, с центром на сайте www.mepis.org. Между прочим, он украсил обложку *LXF79*, где было показано, насколько прост переход на Linux через Mepis. А теперь перед нами новый релиз 6.0, и мощь дистрибутива возрастает с каждой новой версией. Изначально созданный на основе Knoppix, Mepis теперь базируется на Ubuntu, поэтому он сохранил набор основательно протестированных и поддерживаемых пакетов Debian, да еще и получил доступ к более новым пакетам рабочего стола.

Это означает, что SimplyMepis 6.0 соединил лучший выбор пакетов Ubuntu со своими собственными инструментами установки и необходимым для рабочего стола лоском. И более того, Mepis работает в качестве Live-дистрибутива, так что можете испробовать его (или дать другу, ради продвижения Linux) без необходимости устанавливать. Если он вам понравится и вы решите сделать его своим любимым дистрибутивом, можете запустить установщик с рабочего стола – и оп! Пара минут, пара кликов мышью – и вы уже в Mepis.

SimplyMepis специально разработан так, что вам не нужно долго возиться с ним и с добавлением всяких приложений, чтобы он заработал так, как вам хочется.

Дистрибутив включает рабочий стол KDE 3.5.3 с ядром 2.6.15 и *X.org 7.0*, которые и составляют его основу. В нем имеется *OpenOffice.org 2.02* – для работы, а для серфинга – популярный браузер *Firefox*. Чтобы запустить SimplyMepis, загрузите ваш ПК с нашего DVD и войдите в систему как пользователь 'demo' с паролем 'demo'. (если потребуются права суперпользователя, то его пароль – 'root') Ниже приводится краткое руководство по установке SimplyMepis на ваш жесткий диск. Если у вас возникнут проблемы, заходите на www.linuxforum.ru или www.mepis.org/forum. »



› Для настройки системы в SimplyMepis есть простые, но удобные инструменты.



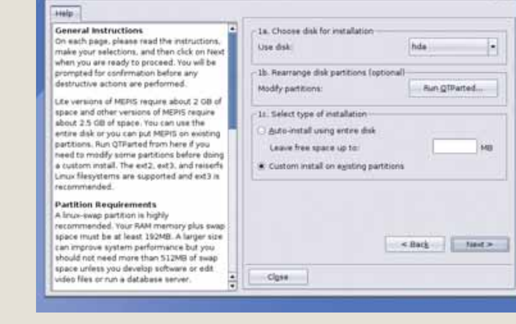
› Рабочий стол для SimplyMepis – фирменная версия KDE 3.5.3.

Шаг за шагом: Установка SimplyMepis на ваш жесткий диск



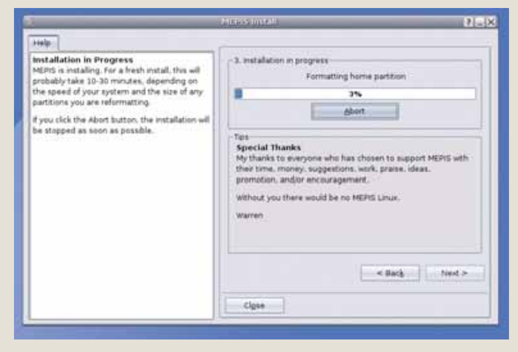
1 Вход

Найдите значок Mepis Install вверху слева на рабочем столе и кликните по нему. В ответ на приглашение, введите пароль администратора ('root'), чтобы начать установку.



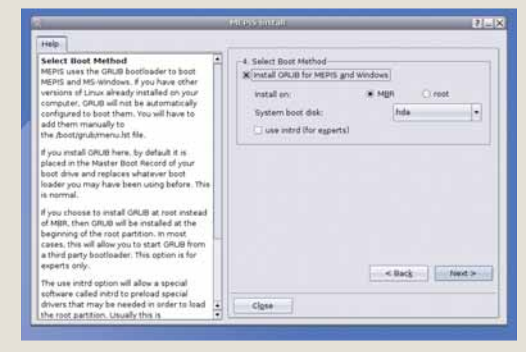
2 Место

В появившемся окне установщика нажмите Next. Можете отдать Mepis все пространство на жестком диске, или выделить раздел, если собираетесь сохранить старую ОС. Прочтите подсказку с левой стороны экрана.



3 Копирование

Установщик начнет копировать файлы на ваш жесткий диск; это займет от 10 до 30 минут. После этого возникнут вопросы по настройке…



4 Перезагрузка

...например, куда установить загрузчик? Можете принять ответы по умолчанию. Наконец, программа сообщит, что надо перезагрузить компьютер, так что извлеките DVD – рестарт, и готово!

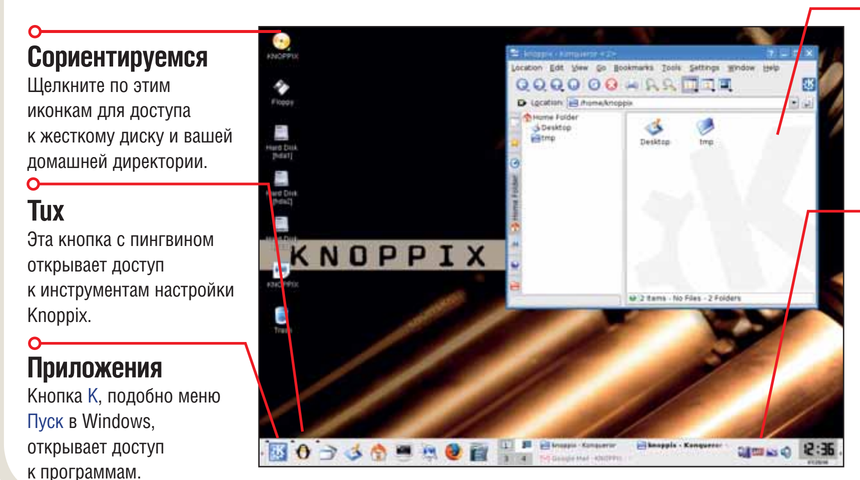
Дистрибутив Linux

Knoppix 5.0

Knoppix был одним из самых популярных и уважаемых Live-дистрибутивов за последние три года и сегодня занимает почетное место в десятке лучших хит-парада DistroWatch.com. Этот дистрибутив был одним пионеров использования Live-дистрибутива как рабочего (а не просто в качестве технической новинки), развивая идею «портативного Linux» и создавая возможность продемонстрировать Linux друзьям и сотрудникам. Благодаря превосходной процедуре распознавания устройств и впечатляющей скорости работы, Knoppix быстро зарекомендовал себя в качестве стандарта де-факто для Live-дистрибутивов Linux.

За последние пару лет появилось множество других Live-дистрибутивов (а некоторые дистрибутивы, например, Ubuntu, обзавелись Live-версией), но Клаус Кноппер [Klaus Knopper], человек, который стоит за Knoppix, не снижает темпов развития. Дистрибутив базируется на тестовой и нестабильной ветвях Debian – так что его пакеты вовсю пользуются результатами тщательной работы сообщества пользователей Debian, предоставляя вам более современный выбор программ, чем постоянные стабильные релизы. Фактически, одним из применений Knoppix можно считать загрузку Debian Testing с последующим обновлением некоторых пакетов (по необходимости).

Исследуем рабочий стол Knoppix



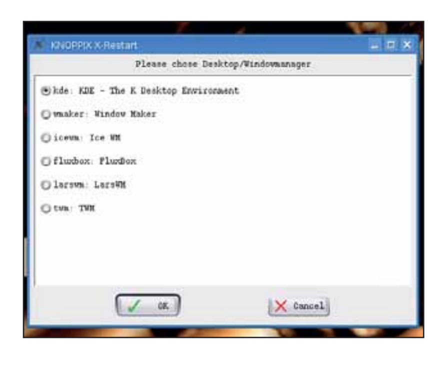
Сориентируемся Щелкните по этим иконкам для доступа к жесткому диску и вашей домашней директории.

Tux Эта кнопка с пингвином открывает доступ к инструментам настройки Knoppix.

Приложения Кнопка K, подобно меню Пуск в Windows, открывает доступ к программам.

Konqueror Файловый менеджер KDE может быть также весьма достойным web-браузером и FTP-клиентом.

Наглядность Щелкните по этому флагу для изменения раскладки клавиатуры. Флаг США подсказывает, что используется американский английский.



› Knoppix предлагает богатый выбор

Версия Knoppix на нашем DVD включает ядро 2.6.17rc, *X.org 7.0* и KDE 3.5.2 в качестве рабочего стола по умолчанию. Поскольку это Live-дистрибутив, можете работать с ним прямо с диска, без установки, а если он вам понравится, можете установить его и на жесткий диск. Рекомендуемые системные требования для работы в Knoppix – процессор 500 Мгц и выше, и как минимум 128 МБ ОЗУ.

Для запуска Knoppix, загрузите ваш ПК с нашего DVD и выберите Knoppix в меню загрузки, используя клавиши управления курсором. Он проанализирует оборудование, запустит некоторые скрипты и все это завершится появлением рабочего стола KDE, хотя вы можете выбрать и другой рабочий стол в начальном меню загрузки – например, менее ресурсоемкий IceWM, который отлично подходит для старых машин. Чтобы установить Knoppix на вашу машину, смотрите приведенную ниже инструкцию, и помните: инсталлятор находится в процессе разработки, так что сначала сделайте резервные копии всех ваших данных, а уж потом приступайте!

Если возникнут проблемы, есть уйма ресурсов, к которым можно обратиться. На диске имеется подробное руководство по установке, а также список часто задаваемых вопросов. Основной форум по Knoppix – www.knoppix.net/forum – посещают тысячи постоянных читателей, которые ответят на все ваши вопросы.

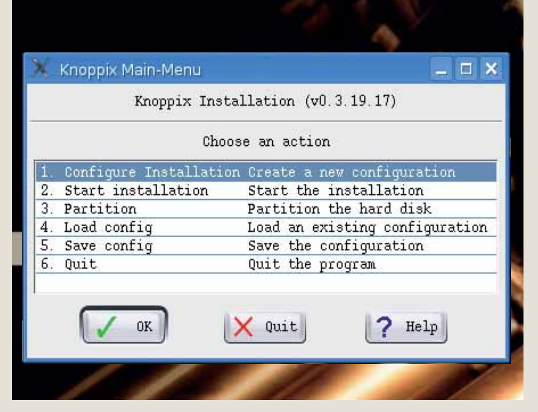
Еще можно посетить www.linuxforum.ru. Удачи!

Шаг за шагом: Установка Knoppix на жесткий диск



1 Запуск инсталлятора

Нажмите на кнопку K, затем Knoppix > Root Shell, и после приглашения войдите в **knoppix-installer**. Когда появится диалоговое окно, нажмите OK, чтобы продолжить установку.



2 Выбор раздела

Если на жестком диске отведен раздел под Linux, на этом экране можно нажать OK и начать установку; если же нет, выберите 3. Partition для выделения необходимого места (мы рекомендуем как минимум 5 ГБ).

Unix-подобная ОС

PC-BSD 1.2

Подумайте об операционной системе на базе BSD, сфокусированной на использовании на рабочих столах, и первое, что придет в голову, будет, очевидно, Mac OS X. Но есть ведь еще и PC-BSD, ответвление FreeBSD – сильный соперник Linux на фронтах борьбы за пользовательские ПК. Если у вас есть LXF77, вы, возможно, помните нашу статью про операционную систему FreeBSD, а если нет, вот вам кратенький обзор ОС и перечисление причин, по которым стоит поработать с ее воплощением для рабочих столов.

FreeBSD – Unix-подобная операционная система с открытым кодом. Появившись в 1993 году, она быстро обзавелась множеством поклонников в серверном мире благодаря своей производительности и стабильности. К несчастью, юридические проблемы, появившиеся на ранней стадии и связанные с кодом, на несколько лет отбросили ее назад, дав шанс народившейся ОС Linux стать основным выбором свободного варианта Unix. Сегодня FreeBSD набрала силу и умеет работать почти со всеми открытыми приложениями для Linux, а специалисты FreeBSD отличаются строгой дисциплиной разработки, благодаря которой количество ошибок удерживается на весьма низком уровне, а время безотказной работы – на высоком.



› На www.pbidir.com находятся автономные двоичные пакеты для работы с PC-BSD.

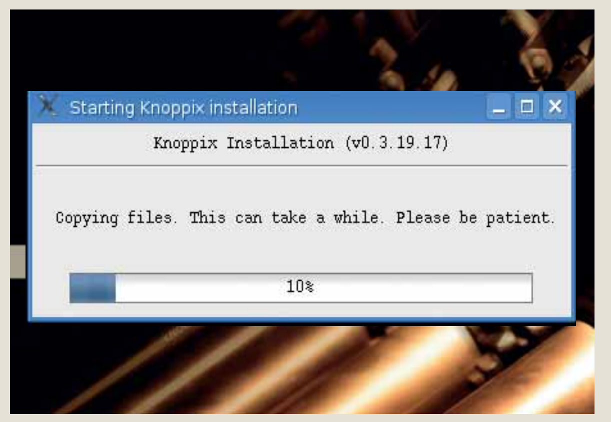


› Вся мощь FreeBSD, но при этом – новый графический инсталлятор и инструменты настройки!

По части настольного применения FreeBSD никогда не была в первых рядах; она, конечно, поддерживает основные устройства, но для ее администрирования нужен опыт работы в Unix. Поэтому Крис Мур [Kris Moore] и его команда взяли FreeBSD, добавили новые инструменты конфигурации и графическую программу установки, привели в порядок рабочий стол и явили миру в качестве PC-BSD – «разработанной с учетом интересов обычного пользователя».

Одним из новшеств является формат пакетов PBI; он включает все зависимости программы (скомпонованные статически), поэтому установка ПО не требует никаких усилий. Сейчас, в версии 1.2, PC-BSD обрела зрелость превосходной операционной системы для настольных ПК, она – идеальный способ расширить горизонт, без отрыва от привычного Linux-окружения. А еще FreeBSD наградила ее впечатляющей стабильностью.

В разделе **Дистрибутивы** (Distros) нашего DVD – два CD-ISO-образа для PC-BSD 1.2; запишите их на CD-R и начните установку, загрузившись с первого диска. (Не забудьте сначала сделать резервные копии данных!) Перед установкой прочтите документацию на диске и посетите www.pc-bsd.org, если потребуется дополнительная информация.



3 Извлечь!

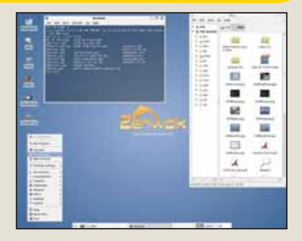
Настроив опции, выберите в меню пункт Start Installation. Начнется копирование файлов. Когда инсталлятор закончит работу, извлеките DVD, и теперь компьютер перезагрузит Knoppix с жесткого диска.

Дистрибутив Linux

Zenwalk Linux

А вы когда-нибудь работали на компьютере в стиле Дзен? Именно этот вопрос задала команда Zenwalk Linux, которая как пчелка трудится над своим компактным дистрибутивом Linux. Основанный на Slackware, Zenwalk (бывший MiniSlack) создан быстрым (он использует легковесную рабочую среду *Xfce*, не заставляя ваш процессор вязнуть в библиотеках Gnome и KDE), рациональным (например, предусмотрена только одна программа на каждую задачу, чтобы не перегружать меню вариантами) и полным. Команда, создавшая дистрибутив, проделала достойную восхищения работу, упаковав солидное количество программ в один CD – а с помощью инструмента управления пакетами *Netpkg* вы можете натаскать кучу полезного из сети.

Zenwalk – прекрасный дистрибутив, чтобы дать вторую жизнь немного устаревшим машинам. Если у вас процессор класса Pentium 3, 128 МБ ОЗУ и 2 ГБ жесткого диска, он будет работать куда быстрее, чем Fedora или SUSE, в большой степени благодаря тому, что рабочим столом по умолчанию является *Xfce*.



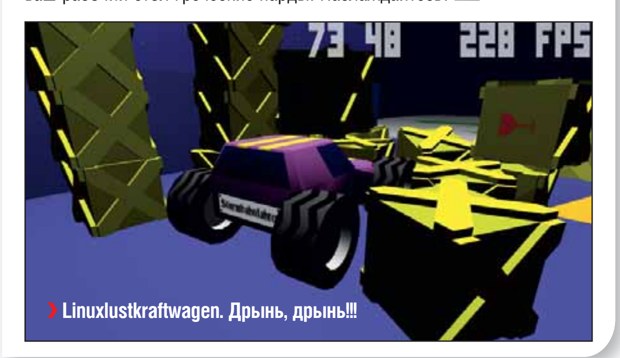
› Для ускорения ритма Zenwalk использует рабочий стол Xfce.

Дополнительным преимуществом является растущее сообщество пользователей Zenwalk, которые уже составили полное руководство пользователя с инструкциями по установке (которое вы найдете на нашем диске в формате PDF).

Для установки Zenwalk, запишите образ ISO из раздела **Distros/Zenwalk** нашего DVD на CD-R, затем загрузите компьютер с вновь записанного диска. (не забудьте, что вы должны записать его как образ диска, а не просто скопировать файл. Вам, возможно, понадобится выставить в настройках BIOS загрузку с CD-ROM, а не с HDD). Если при установке возникнут проблемы, зайдите на сайт на www.zenwalk.org, где вы найдете полезные форумы, на которых вам помогут другие пользователи Zenwalk.

И наконец...

Несколько кратких заметок по поводу прочих изюминок DVD. В раздел **Интернет** мы включили *Opera*, быстродействующий браузер, рассмотренный на стр.10, а в раздел **Рабочий стол** поместили несколько удаленных клиентов рабочего стола, о которых писали в Сравнении. Любители загадочного могут насладиться Minix 3, маленькой Unix-подобной ОС, с которой начинал Торвальдс до того, как приступить к написанию собственного ядра. Ну, и чтобы расслабиться после беспокойного дня, загляните в раздел **Игры**, где вы найдете новые релизы развлекательных программ. *Sturmbahnfahrer* – потрясающий тренажер вождения, в котором имеются маршруты с препятствиями; *Gnome Hearts* – классическая карточная игра; *Tavli 3d* поставит на ваш рабочий стол греческие нарды. Наслаждайтесь! LXF



› Linuxlustkraftwagen. Дрынь, дрынь!!!



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
Литературные редакторы
Родион Водейко, Иван Мищенко, Елена Толстякова
Переводчики
Александр Бикмеев, Светлана Кривошеина, Александр Кузьменков, Алексей Опарин, Сергей Супрунов, Александр Черных, Юлия Шабунио.
Допечатная подготовка
Мария Пучкова, Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Павел Фролов, Станислав Медведев

Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____
Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейч (Nick Veitch) nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk
Старший художественный редактор Мартин Парфитт (Martin Parfitt) mparfitt @futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса (Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk
Новостной редактор Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk
Литературный редактор
Ребекка Смолли (Rebecca Smalley) rebecca.smalley@futurenet.co.uk
Штатный автор
Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk
Ассистент по выпуску
Эндрю Грегори (Andrew Gregory) andrew.gregory@futurenet.co.uk

Авторы
Марк Бейн (Mark Bain), Ладислав Боднар (Ladislav Bodnar), Нейл Ботвик (Neil Bothwick), Д-р. Крис Браун (Dr. Chris Brown), Энди Ченнел (Andy Channelle), Ричард Драммонд (Richard Drummond), Дэвид Кулсон (David Coulson), Евгений Балдин, Андрей Боровский, Андрей Дубравин, Дмитрий Кирсанов, Сергей Супрунов, Тихон Тарнавский, Антон Черноусов
Подготовка к печати: Сэлли Хейлз (Sally Hales), Беки Скьюз (Becky Skuse)
Дизайн: Зигги Бейкер (Ziggi Baker), Анна Фишер (Anna Fisher), Эмит Пэйтел (Amit Patel)
Фотографии: Энди Шот (Andy Short)
Иллюстрации: Мэт Херинг (Matt Herring), Энди МакЛафлин (Andy McLaughlin), Magictorch, Джеми Слиммон (Jamie Slimmon), Отто Штейнингер (Otto Steininger), Алекс Вильямсон (Alex Williamson)

КОНТАКТНАЯ ИНФОРМАЦИЯ
England: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email linuxformat@futurenet.co.uk
Россия: Санкт-Петербург, ул. Гончарная, 23, офис 54,
телефон: (812) 717-00-37
Email: info@linuxformat.ru, Web: www.linuxformat.ru



В ноябре

Novell vs Red Hat

SUSE Linux Enterprise 10 уже здесь: давайте разберемся, как SUSE продвигает Linux на предприятия и что это значит для Red Hat



Гаэль Дюваль возвращается

Основатель Mandrake Linux вернулся с новым дистрибутивом. Встречайте – Ulteo!

Проблемы настольного Linux

Почему Пол Хадсон боится Gnome 3.0



Интервью LXF

Новости Syllable – из первых рук

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА В ЛИНУКСЦЕНТРЕ

Сколько стоит подписка?

Подписка на журнал «Linux Format» 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь) стоит 1800 рублей
Подписка на журнал «Linux Format» 6 номеров (июль, август, сентябрь, октябрь, ноябрь, декабрь 2006 года) стоит 900 рублей

Как оформить подписку?

Чтобы оформить подписку на журнал «Linux Format», необходимо зарегистрироваться в интернет-магазине Linuxcenter.Ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар «Подписка на журнал «Linux Format» 12 номеров 2006 года», или товар «Подписка на журнал «Linux Format» второе полугодие 2006 года», получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Как оплатить подписку?

– по выставленному счету (для юридических лиц)
– по квитанции в любом отделении Сбербанка

Плюсы подписки

- подписка дешевле!
- гарантированное получение нового номера журнала!

ПОДПИСКА - 2007!

ПОДПИСКА ПО КАТАЛОГАМ РФ

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882

Каталог «ПРЕССА РОССИИ» – подписной индекс 87974

Ф. СП-1

Министерство связи РФ

АБОНЕМЕНТ НА ЖУРНАЛ

Linux Format

ИНДЕКС ИЗДАНИЯ

КОЛИЧЕСТВО КОМПЛЕКТОВ

НА 2007 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА ПОЧТОВЫЙ ИНДЕКС АДРЕС ДОСТАВКИ

КОМУ ФАМИЛИЯ, ИНИЦИАЛЫ

ДОСТАВОЧНАЯ КАРТОЧКА

ПВ МЕСТО ЛИТЕР

НА ЖУРНАЛ ИНДЕКС ИЗДАНИЯ

Linux Format

НАИМЕНОВАНИЕ ИЗДАНИЯ

СТОИМОСТЬ: ПОДПИСКИ ___ РУБ. ___ КОП.; ПЕРЕАДРЕСАЦИИ ___ РУБ. ___ КОП.; КОЛИЧЕСТВО КОМПЛЕКТОВ

НА 2007 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА ПОЧТОВЫЙ ИНДЕКС АДРЕС ДОСТАВКИ

КОМУ ФАМИЛИЯ, ИНИЦИАЛЫ



ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА ПО КАТАЛОГАМ СНГ И БЛИЖНЕГО ЗАРУБЕЖЬЯ

Каталог «Российская Пресса» – совместный проект Государственного предприятия «Казпочта», Агентства «Книга-Сервис» и АРЗИ.

Блок изданий АРЗИ в национальных Каталогах Украины и Беларуси. В Азербайджане, Армении, Грузии, Киргизии, Узбекистане и Молдове - по изданиям, включенным в Объединенный каталог, распространяемые через АРЗИ.

Азербайджан

- по Объединенному каталогу российских изданий через Предприятие по распространению печати «Гасид»(370102, г. Баку, ул. Джавадхана, 21);

Армения

- по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г.Ереван, пл.Сасунци Давида, д.2) и ЗАО «Контакт-Мамул» (375002, Г.Ереван, ул.Сарьяна, 22);

Белоруссия

- по Каталогу изданий стран СНГ через РГО «Белпочта» (220050, г.Минск, пр-т Ф.Скорины, 10);

Грузия

- по списку номенклатуры «АРЗИ»» через АО «Сакпресса» (380019, г.Тбилиси, ул.Хошараульская, 29) и АО «Мацне» (380060, г.Тбилиси, пр-т Гамсахурдия, 42);

Казахстан

- по Каталогу «Российская Пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»;

Молдавия

- по каталогу через ГП «Пошта Молдавей» (МД-2012, г.Кишинев, бул.Штефан чел Маре, 134);
- по списку через ГУП «Почта Приднестровья» (MD-3300, г.Тирасполь, ул.Ленина, 17);
- по прайс-листу через ООО Агентство «Editil Periodice» (2012, г.Кишинев, бул. Штефан чел Маре, 134).

Узбекистан

- по Каталогу «Davriy nashrlar» российские издания через Агентство по распространению печати «Davriy nashrlar» (7000029, Ташкент, пл.Мустакиллик, 5/3, офис 33);

Украина

- Киевский главпочтамт.
- Подписное агентство «KSS» Телефон/факс (044)270-62-20, 270-62-22

ПОДПИСКА НА LINUX FORMAT

Агентство “Centerpress”

Сколько стоит подписка?
Подписка на журнал "Linux Format" 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь 2007 года) стоит 1800 рублей.

Как оформить подписку?
Чтобы оформить подписку на журнал "Linux Format", необходимо зарегистрироваться в интернет-агентстве Centerpress.ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар "Подписка на журнал "Linux Format" на 2007 год 12 номеров (01-12 / 2007)", получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Агентство “Centerpress”: www.centerpress.ru

Все Плюсы подписки!
- Подписка дешевле!
- Гарантированное получение журнала!

По каталогам РФ

Каталог агентства "РОСПЕЧАТЬ" - подписной индекс

20882

Каталог "ПРЕССА РОССИИ" - подписной индекс

87974



АЛЬТЕРНАТИВНЫЕ АГЕНСТВА РФ

Агентство «Интер-Почта» (095) 500-00-60, курьерская доставка по Москве.

Агентство «Вся Пресса» (095) 787-34-47

Агентство «УралПресс»

- Екатеринбург, Березовский, В. Пышма, Первоуральск
  тел. (343) 375-80-71, 375-84-93, 375-84-39, факс 375-62-74, info@ural-press.ru
- Нижний Тагил
  тел. (3435) 411448, 417709, ntagil@ural-press.ru
- Челябинск
  тел. (351) 262-90-03, 262-90-05, pochta@chel.surnet.ru
- Пермь
  тел. (3422) 60-24-40, 60-22-95, 60-35-42, parma-press@permonline.ru